

GESTIÓN DE TEMAS Y MEJORAS EN ACIDE THEME MANAGEMENT AND ENHANCEMENTS FOR ACIDE



TRABAJO FIN DE GRADO
CURSO 2020-2021

AUTOR
CARLOS GONZÁLEZ TORRES

DIRECTOR
PROF. FERNANDO SÁENZ PÉREZ

GRADO EN INGENIERÍA DEL SOFTWARE
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

Resumen

GESTIÓN DE TEMAS Y MEJORAS EN ACIDE

ACIDE (A configurable IDE) es un entorno de desarrollo gráfico de código abierto, integrado, gratuito y multiplataforma, que se puede configurar para ser utilizado con cualquier sistema de desarrollo, como intérpretes, compiladores y sistemas de bases de datos. En particular, ACIDE puede ser configurado como interfaz gráfica para DES (Datalog Educational System)[13].

DES [13] permite depurar bases de datos a través de una interfaz por consola, de ahí que resulte mucho más intuitivo integrarlo con ACIDE y aprovechar la interfaz gráfica y funcionalidades extras que nos ofrece. Es un sistema de bases de datos deductivo con Datalog, SQL, Álgebra Relacional (RA), Cálculo Relacional de Tuplas (TRC) y Cálculo Relacional de Dominio (DRC) como lenguajes de consulta.

En la versión ACIDE que me ocupa se han añadido nuevas funcionalidades a la aplicación que mejoran la experiencia de usuario, entre las que destacan: la gestión de temas, que posibilita cambiar la distribución de colores para la aplicación y sus fuentes, y poder guardarlo; y la configuración de codificación a los archivos cargados, que permite seleccionar con qué codificación mostrar los ficheros abiertos en la aplicación. Además, se han implementado muchas otras características y realizado tareas de mejora y subsanación de errores.

Palabras clave

ACIDE, DES, IDE, temas, codificación, consola, editor de archivos, formato, léxico.

ABSTRACT

THEME MANAGEMENT AND ENHANCEMENTS FOR ACIDE

ACIDE (A configurable IDE) is an open source, integrated, free, cross-platform, configurable graphical development environment that can be configured for use with any development system, such as interpreters, compilers and database systems. In particular, ACIDE can be configured as a graphical interface to DES (Datalog Educational System) [13].

DES [13] allows debugging databases through a console interface, which makes it much more intuitive to integrate it with ACIDE and take advantage of the graphical interface and extra functionalities it offers. It is a deductive database system with Datalog, SQL, Relational Algebra (RA), Tuple Relational Calculus (TRC) and Domain Relational Calculus (DRC) as query languages.

In this ACIDE version new functionalities have been added to the application that improve the user experience, among which the following stand out: theme management, which makes it possible to change the colour layout for the application and its fonts, and to save it; and the coding of the loaded files, which makes it possible to select the coding with which to display the files opened in the application. In addition, many other features have been implemented and improvements and bug fixes have been made.

Keywords

ACIDE, DES, IDE, themes, encoding, console, file editor, format, lexicon.

ÍNDICE DE CONTENIDOS

Resumen.....	2
Abstract.....	3
Índice de contenidos	4
Índice de figuras	7
Capítulo 1 - Introducción	10
1.1 Motivación	10
1.2 Objetivos.....	11
1.2.1 Objetivos no alcanzados.....	11
1.3 Antecedentes del trabajo	13
1.3.1 Evolución de ACIDE	13
1.4 Plan de trabajo	17
Capítulo 2 - Gestión de los temas en ACIDE.....	20
2.1 Diagrama de clases para la funcionalidad principal	29
Capítulo 3 - Otras implementaciones y mejoras	33
3.1 Añadir opciones de codificación	33
3.2 Permitir mantener las dimensiones de ACIDE.....	37
3.3 Configurar el botón ejecutar de ACIDE	37
3.4 Redimensión número de reglas Asserted database	39
3.5 Permitir pegar contenido en la consola en la posición del cursor.....	40
3.6 Corrección del comportamiento de ACIDE al cerrar un proyecto	41
3.7 Creación de un instalador para ACIDE.....	42
3.8 Crear tabla con Datalog	44

3.9 Corrección de la opción de ajuste de línea	46
3.10 Cambio de formato de archivos	47
3.11 Otras correcciones o mejoras.....	49
Capítulo 4 - Conclusiones y trabajo futuro	51
4.1 Conclusiones	51
4.2 Trabajo futuro	51
Chapter - Introduction	53
5.1 Motivation	53
5.2 Objectives	54
5.2.1 Objectives not achieved.....	54
5.3 Background of the project	55
5.3.1 Development of ACIDE	55
5.3.1.1 Source code standards	57
5.4 Work plan	58
Chapter - Conclusions and future work.....	59
6.1 Conclusions	59
6.2 Future work	59
Bibliografía.....	61
Apéndices	63
7.1 Introducción.....	63
7.2 Requisitos previos	63
7.3 Proceso de creación.....	64
7.3.1 Generación de archivo jar.....	64
7.3.2 Generación del ejecutable .exe:	67

7.3.3 Creación del instalador:	71
--------------------------------------	----

ÍNDICE DE FIGURAS

Figura 1-1 Comportamiento correcto de la consola	12
Figura 1-2 Error consola al iniciar ACIDE	12
Figura 1-3 Código de licencia ACIDE	16
Figura 1-4 Código Javadoc de las clases	16
Figura 1-5 Estándar para el nombre de las clases	16
Figura 1-6 Formato para las variables constantes	17
Figura 1-7 Métodos obligatorios para nuevas ventanas.....	17
Figura 1-8 Web de ACIDE	19
Figura 2-1 Vista del menú configuración.....	20
Figura 2-2 Menú desplegable temas	21
Figura 2-3 Vista para configurar nuevos temas	22
Figura 2-4 Configuración de color de fuente	22
Figura 2-5 Configuración de colores por código HSV	23
Figura 2-6 Panel vista previa configuración de colores	24
Figura 2-7 Diálogo para confirmar el salvaguardado del tema	25
Figura 2-8 Tema registrado en el sistema ACIDE.....	25
Figura 2-9 Configuración de léxico tema claro	26
Figura 2-10 Configuración de léxico tema oscuro	27
Figura 2-11 Opciones para cada tema.....	27
Figura 2-12 Panel edición de tema.....	28
Figura 2-13 Estructura archivo properties de un tema	28
Figura 2-14 Clases afectadas al añadir el menú temas.....	30

Figura 2-15 Listener que asignan el comportamiento del acceso al menú temas	31
Figura 2-16 Clase de la ventana para la configuración de temas.....	32
Figura 3-1 Opciones de codificación	34
Figura 3-2 Codificaciones disponibles	34
Figura 3-3 Codificación por defecto windows-1252	35
Figura 3-4 Codificación UTF-8 reconocida	36
Figura 3-5 Barra tareas con codificación activa	36
Figura 3-6 Opción para ocultar botón ejecutar ACIDE	38
Figura 3-7 Botón ejecutar disponible	38
Figura 3-8 Barra de herramientas con botón de ejecutar oculto	39
Figura 3-9 Botón de ejecutar oculto	39
Figura 3-10 Campo reglas en Base de datos acertada	40
Figura 3-11 Menú popup de la consola. Accesible con el botón derecho del ratón	41
Figura 3-12 Vista ACIDE sin proyecto activo	42
Figura 3-13 Script de creación del instalador	44
Figura 3-14 Error diálogo crear tabla con Datalog.....	44
Figura 3-15 Diálogo actual para crear tabla con Datalog	45
Figura 3-16 Menú desplegable Tablas.....	45
Figura 3-17 Error opción ajuste de línea, no lleva el foco del texto al final	46
Figura 3-18 Opción de ajuste de línea activa.....	46
Figura 3-19 Opción ajuste de línea desactivada	47
Figura 3-20 Menú formato de archivos.....	48
Figura 3-21 Formato DOS seleccionado en ACIDE.....	49
Figura 3-22 Proyecto guardado con el mismo nombre	50

Figura 3-23 Mensaje para confirmar la sobreescritura del proyecto	50
Figura 4-1 Estructura necesaria de la carpeta donde se aloje el bat	84
Figura 4-2 Contenido carpeta Files.....	85
Figura 4-3 Contenido de la carpeta Folders	85
Figura 4-4 Sección Files donde registrar todos los archivos que se quieran añadir.....	86
Figura 4-5 Sección para borrar cualquier archivo de ACIDE al desinstalar	86
Figura 4-6 Contenido archivo bat.....	86

Capítulo 1 - Introducción

En esta memoria se describe la implementación de la gestión de los temas en ACIDE como tarea principal del trabajo de final de grado. Además, se describirá el resto de las funcionalidades que se han desarrollado, contribuyendo a una mejora del funcionamiento de la aplicación, así como de la experiencia de usuario.

En los respectivos apartados se explicará de manera detallada cada una de las funcionalidades implementadas. Al final del documento se encuentra una guía explicativa del proceso de generación del instalador de la aplicación.

1.1 Motivación

Durante los años que he estado cursando la titulación de ingeniería del software siempre he tenido clara mi vocación por el desarrollo de aplicaciones y programas, que se ha visto acentuada con el paso del tiempo. ACIDE fue la primera opción que valoré, al tratarse de un proyecto que responde plenamente a mis inquietudes.

La finalidad de este trabajo es dotar a ACIDE de características que le otorguen una mejor experiencia de usuario, así como de funcionamiento. Al no estar enfocadas en un tema concreto, las funcionalidades desarrolladas han sido enmarcadas bajo tareas prioritarias pendientes de realización.

La carencia de sistemas que permitan la depuración de bases de datos SQL motiva el desarrollo de ACIDE. DES [13] nos permite realizar una depuración textual, que unida a la herramienta de interfaz gráfica que ofrece ACIDE para la programación y depuración de bases de datos, hace necesario el avance continuo en el proceso de mejora de la aplicación, consolidándose como una herramienta de gran ayuda para eventuales programadores. Con el fin aumentar la usabilidad del programa y seguir creando una interfaz amigable para el usuario, se ha proseguido con el proceso de mejora y adición de nuevas funcionalidades en ACIDE.

1.2 Objetivos

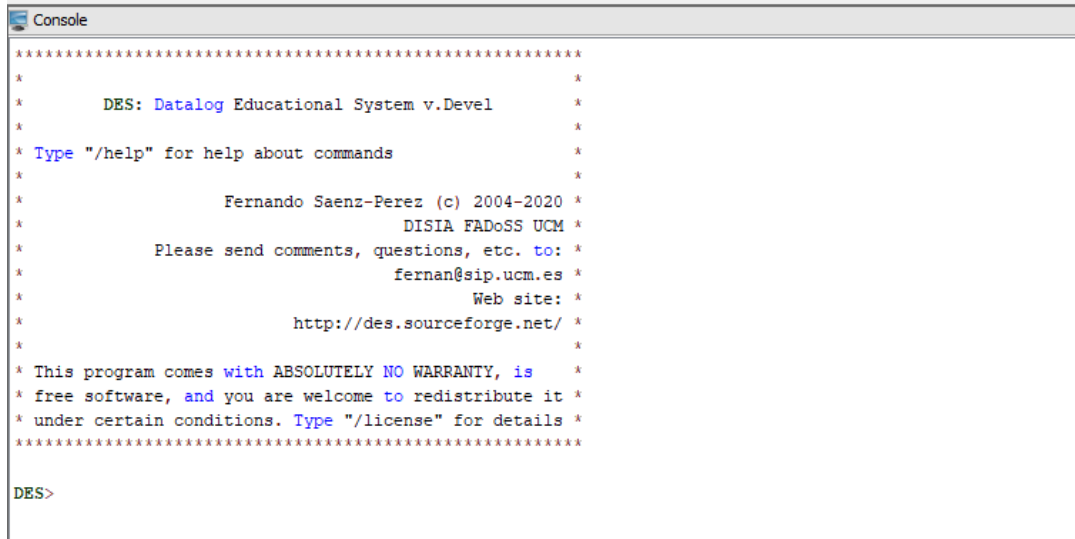
Este trabajo de final de grado tiene como objetivos la implementación de diversas funcionalidades pendientes de desarrollo y la corrección o mejora de algunas de las existentes.

Como funcionalidad principal se ha desarrollado un sistema de gestión de temas que permite configurar ACIDE con la gama de colores para los fondos y texto de paneles y menús que se deseen. Además, se permite guardar dicha configuración como un tema de la aplicación y poder aplicarlo cuando se desee. El funcionamiento se detallará de manera extensa en apartados posteriores.

También se han implementado muchas otras características que se mencionarán más adelante. Por nombrar algunas: la posibilidad de mostrar los archivos cargados a la aplicación bajo las codificaciones que ofrece; la redimensión de los paneles y ventanas de la aplicación y la posibilidad de hacerlo persistente al salir de la misma; permitir la ocultación de los submenús de la aplicación desde el menú de configuración de ACIDE; vinculación del léxico a cada tema creado; y añadir la opción ayuda de DES [13] que abre un archivo con el manual.

1.2.1 *Objetivos no alcanzados*

En el desarrollo de este proyecto no ha sido posible subsanar el error que se produce al iniciar ACIDE: la consola no se inicia de manera correcta. El comportamiento esperado debe ser el que se refleja en la siguiente captura.



```
*****
*
*      DES: Datalog Educational System v.Devel
*
*      Type "/help" for help about commands
*
*      Fernando Saenz-Perez (c) 2004-2020
*      DISIA FADoSS UCM
*      Please send comments, questions, etc. to:
*      fernan@sip.ucm.es
*      Web site:
*      http://des.sourceforge.net/
*
* This program comes with ABSOLUTELY NO WARRANTY, is
* free software, and you are welcome to redistribute it
* under certain conditions. Type "/license" for details
*****
DES>
```

Figura 1-1 Comportamiento correcto de la consola

Al iniciar ACIDE de manera aleatoria, sin un patrón constante, aparece la orden \$success, debiendo ser reiniciada la consola para que muestre el resultado de la captura anterior.



```
Console
$success
```

Figura 1-2 Error consola al iniciar ACIDE

Durante el tiempo invertido en intentar subsanar este fallo, se ha identificado el error procedente del orden de ejecución de los hilos que crea ACIDE al iniciarse. Sin

embargo, no ha sido posible delimitar qué invocación de los comandos que ejecuta cada uno de ellos provoca esta salida.

1.3 Antecedentes del trabajo

ACIDE es un entorno de desarrollo integrado que puede ser utilizado fácilmente como intérprete, compilador o sistema de bases de datos para Álgebra Relacional, TRC, DRC, Datalog y SQL.

Este proyecto se constituye como un trabajo de evolución de la herramienta, que ha sido desarrollada anteriormente por otros grupos:

- El primer año que se llevó a cabo fue en el curso académico 2006-2007, por Diego Cardiel Freire, Juan José Ortiz Sánchez y Delfín Rupérez Cañas.
- Durante el curso académico 2007-2008 el desarrollo fue continuado por Miguel Martín Lázaro.
- La siguiente iteración en el desarrollo de ACIDE fue en el curso 2010-2011 por Javier Salcedo Gómez.
- Durante el curso académico 2012-2013 el desarrollo de ACIDE correspondió a Pablo Gutiérrez García-Pardo, Elena Tejeiro Pérez de Ágreda y Andrés Vicente del Cura.
- En el curso académico 2013-2014 el desarrollo se realizó por Juan Jesús Marqués Ortiz, Fernando Ordás Lorente y Semíramis Gutiérrez Quintana.
- Durante el curso 2014-2015 el desarrollo se realizó por Sergio Domínguez Fuentes.
- Y durante el curso 2019-2020 el desarrollo lo efectuó Sergio García Rodríguez.

Este proyecto siempre ha estado dirigido por el profesor Fernando Sáenz Pérez.

1.3.1 Evolución de ACIDE

Para valorar la magnitud de ACIDE a continuación se detalla la evolución del proyecto desde sus inicios hasta la versión actual.

ACIDE nace de la necesidad de crear un IDE configurable para distintos lenguajes de programación, que ofrezca al usuario una herramienta gráfica con multitud de

opciones inteligibles. Como se declara en las primeras memorias dedicadas al desarrollo de ACIDE, se tomaron como referencia algunos editores de texto, tales como Crimson Editor [14] y JEdit [15].

Continuando con el planteamiento de ACIDE, se identificó qué tipo de entorno de desarrollo se debía crear para cumplir con la idea inicial de la aplicación. Al tratarse de un IDE pensado para diferentes lenguajes de programación, se tomaron como referencia entornos no especializados en un único lenguaje, en este caso Eclipse [19]. Sin embargo, siendo una aplicación que limita sus herramientas a los plugins ofrecidos por sus desarrolladores, y en ocasiones tediosa de usar, se valoraron otras enfocadas a lenguajes concretos que pudiesen ofrecer características interesantes, para facilitar lo máximo posible el uso de ACIDE. En este grupo se mencionan programas como: JBuilder [16], JCreator [17] y C++ Builder [18].

En la siguiente evolución de ACIDE se tomó como referencia Visual Studio Shell [20], que posibilita la adaptación de la aplicación a un lenguaje propio y la creación de herramientas personalizadas.

Más tarde, se añadieron las aplicaciones WinEdt [21] y NetBeans [22] como referencias para el desarrollo de ACIDE, además de seguir teniendo en cuenta las anteriores.

Durante el posterior desarrollo de ACIDE surgió la necesidad de vincular la consola con ODBC y DES [13]; para ello se tomaron como referencia los motores gráficos de los sistemas gestores de bases de datos MS Access [23], Oracle [24] y Postgres [25]. Además, se comenzó a tener en cuenta ASPIDE [26], con funcionalidades similares a ACIDE.

La evolución prosiguió con la implementación de diversas funcionalidades aprovechando la vinculación existente con DES [13] y el sistema ODBC. Se crearon nuevos paneles que dotasen a ACIDE de más opciones, así como tareas de mejora para corregir o mejorar las ya existentes. Muchas de las funcionalidades que ofrecía ACIDE en ese momento se basaron en el editor ya mencionado, Crimson Editor [14].

A continuación, se incidió en el deseo de aumentar la funcionalidad del proyecto, agregando la traducción al francés y creando un nuevo panel que sirviese para la futura depuración. Se llevaron a cabo también tareas de mejora de las opciones ya disponibles, para simplificar su uso al máximo y dotar ACIDE de una interfaz amigable para el usuario.

Finalmente, a la versión desde la que partí se le había añadido una ampliación de la depuración presente en la última versión de DES [13] disponible, que permitía configurar el estado de tablas y vistas que no están seleccionadas, mejorando la usabilidad del sistema, además de tareas de mantenimiento y corrección de errores en ACIDE.

1.3.1.1 Estándares de código fuente

Al tratarse de un proyecto en desarrollo por múltiples desarrolladores, ha sido necesario implantar unos estándares de código. Su finalidad es poder implementar las nuevas funcionalidades o mejoras bajo un mismo esquema de programación, que facilite en gran medida la futura adición de nuevas opciones o la mantenibilidad de ACIDE.

En primer lugar, todos los comentarios e identificadores han sido declarados en inglés. Además, al inicio de las clases presentes en el proyecto se indica la licencia que presenta, GPLv3.

```

/*
This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

```

Figura 1-3 Código de licencia ACIDE

En segundo lugar, se han introducido comentarios en cada una de las funciones y atributos que se han añadido al código, con el fin de indicar cuál es su propósito y la funcionalidad que desempeñan.

Asimismo, cada una de las clases Java existentes en el proyecto dispone de un código Javadoc con el siguiente formato:

```

/**
 * Descripción de la clase.
 *
 * @version 0.11
 * (@see <NombreDeClase/NombreDeInterfaz>)
 */

```

Figura 1-4 Código Javadoc de las clases

En tercer lugar, todas las variables declaradas en las clases en ACIDE comienzan por "_", con el fin de homogeneizar la nomenclatura de los atributos que se declaran. Además, la denominación de las clases comienza por Acide, seguido de un nombre descriptivo de su funcionalidad. Todas las palabras que conformen el nombre empezarán por mayúscula, respetando de esta manera el estándar de Java.

```

public Class AcideMenuBar extends JMenuBar {...}

```

Figura 1-5 Estándar para el nombre de las clases

Los nombres de los métodos declarados adoptan el mismo formato descrito para las clases.

En cuarto lugar, cuando sea necesario declarar variables constantes en el código, su nombre debe ir en mayúscula; si está representado por varias palabras, se separarán haciendo uso de “_”.

```
public static final String COMPILER_NAME;  
public static final ImageIcon COMPILER_IMAGE;  
  
//Builds the ACIDE - A Configurable IDE configuration window  
//components  
private void buildComponents() {...}  
//Adds the components to the ACIDE - A Configurable IDE to the  
//configuration window  
private void addComponents() {...}  
//Sets the text of the ACIDE - A Configurable IDE class components  
//with the labels in the selected language to display  
private void setTextOfMenuComponents() {...}  
//Updates the ACIDE - A Configurable IDE class components  
//visibility with the menu configuration  
private void updateComponentsVisibility() {...}  
//Sets the listeners of the configuration window components.  
private void setListeners() {...}
```

Figura 1-7 Métodos obligatorios para nuevas ventanas

Y en quinto y último lugar, todas las clases que hacen referencia a ventanas de la aplicación necesariamente deberán incluir los siguientes métodos:


1.4 Plan de trabajo


Al comienzo del proyecto, su director, el profesor Fernando Sáenz Pérez, me facilitó el acceso a una carpeta compartida en Google Drive [3] donde poder acceder al archivo comprimido que contenía las múltiples carpetas y archivos indispensables para ejecutar la última versión de ACIDE; su código fuente sobre el que empezar a trabajar; así como una relación priorizada de las tareas pendientes a realizar.



En base a esto, comencé a desarrollar las funciones o características siguiendo el orden de prioridad estipulado en dicho documento. Cada semana comunicaba al director el estado del proyecto y los avances realizados.

Una vez tenía funcionalidades acabadas, generaba un ejecutable de la aplicación para enviárselo al director para su posterior validación. De esta manera, he ido mejorando y ampliando las características de ACIDE hasta conseguir la versión actual del sistema (ACIDE 0.19). Todas las tareas completadas están alojadas en un repositorio de Github, creado para mantener un control de las versiones que he ido implementando. (<https://github.com/Cargon16/AcideTFG>)

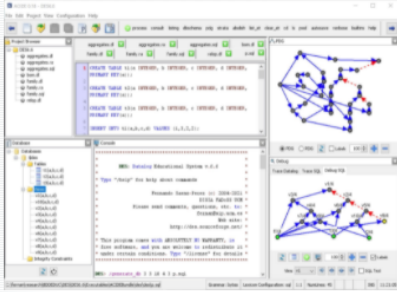
Como comentario adicional, dejar constancia de que, tras desarrollar varias mejoras o funciones, se procedió a publicar otra versión intermedia a la actual (ACIDE 0.18.1). Esta versión de la aplicación está disponible en la web dedicada a ACIDE (<http://www.fdi.ucm.es/profesor/fernand/ACIDE/>). En ella se pueden encontrar las versiones de ACIDE configuradas para cada uno de los sistemas operativos (Windows, Linux y Mac), un manual de usuario, un archivo con los cambios implementados en la última versión y diversas secciones de la web que ofrecen diversas opciones, tales como: capturas de la aplicación, información sobre los desarrolladores que hemos contribuido, estadísticas de las descargas de ACIDE y paquetes preconstruidos.





[Download](#)



[Home](#)
[Download](#)
[Screenshots](#)
[Release Notes](#)
[License](#)
[News](#)
[Contact](#)
[Credits](#)
[Publications](#)
[Statistics](#)
[Pre-built Bundles](#)




NEWS:

March 28th, 2021:

ACIDE 0.18.1 with some enhancements, fixes and prebuilt bundles

January 25th, 2021:

ACIDE 0.18 for DES including a graphical debugger



This project is aimed to provide a free and cross-platform configurable graphical integrated development environment which can be configured in order to be used with any development system such as interpreters, compilers and database systems.

Features of this system include: Project management, Multifile editing, Syntax colouring, Console panel, and Database panel (tested with DB2, MySQL, Oracle, PostgreSQL, SQL Anywhere and DES).

End-users who can benefit from this system include: console and database users, researchers that develop programming systems, and developers. It is implemented in Java in order to be platform independent.

The current release adds many new features and fixes bugs, but it remains to fix many others, remarkably the parsing-on-the-fly.

This tool is partially described in: "[ACIDE: An Integrated Development Environment Configurable for LaTeX](#)", *PracTeX Journal* 2007-3, special issue on "Tools for LaTeX and TeX Users".

Web site last update: March 28, 2021

Figura 1-8 Web de ACIDE

Capítulo 2 - Gestión de los temas en ACIDE

Según he indicado en los puntos anteriores, esta ha sido la característica principal que se ha desarrollado en este trabajo de fin de grado. El objetivo era dotar a la aplicación de un sistema que permitiese al usuario personalizar las vistas disponibles con los colores de fuente y fondo que considerase oportunos.

En este punto desarrollaré de manera extensa los elementos necesarios para su implementación, así como el modo de uso y todas las opciones que nos ofrece.

Primeramente, para poder acceder a todas las opciones se debió crear el acceso correspondiente en el menú de la aplicación, en concreto, al tratarse de una modificación en su configuración, decidí incluirlo en la opción que lleva el mismo nombre.

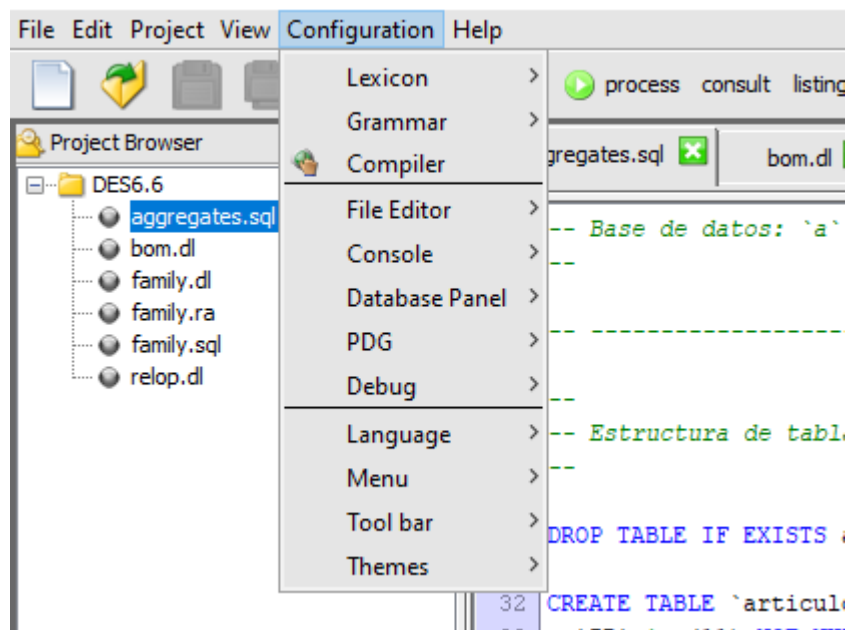


Figura 2-1 Vista del menú configuración

Para mantener el mismo patrón que seguían el resto de los elementos que se encuentran en ese menú y facilitar el entendimiento al futuro usuario, decidí incluir un menú que desplegase todas las opciones que nos ofrece esta funcionalidad.

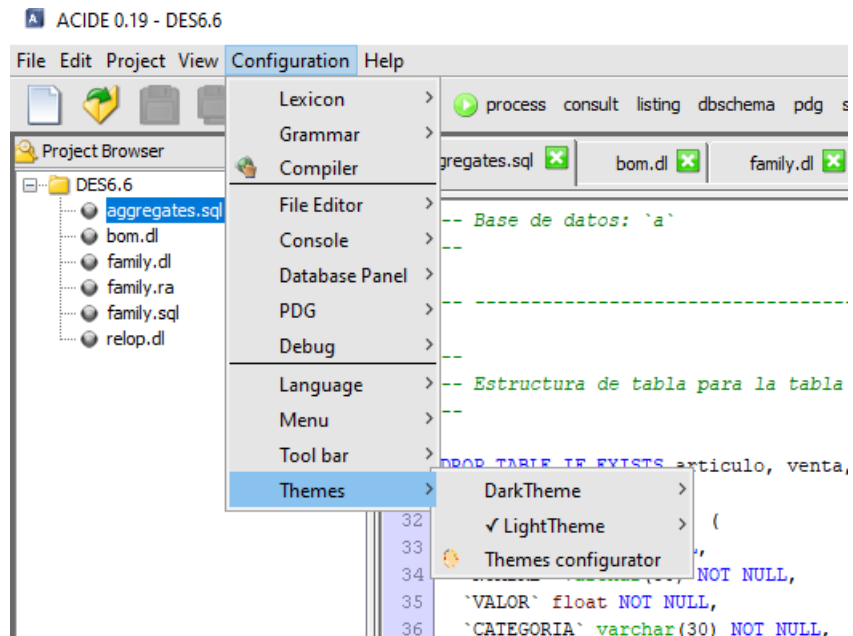


Figura 2-2 Menú desplegable temas

Como se puede observar en la captura anterior, al abrir el menú, nos aparecen los temas creados de manera predeterminada en la aplicación, así como un acceso que nos mostrará una ventana donde configurar un tema nuevo.

Las opciones que se despliegan para cada tema se explicarán más adelante. A continuación, se detalla la manera de proceder para la configuración de un tema nuevo.

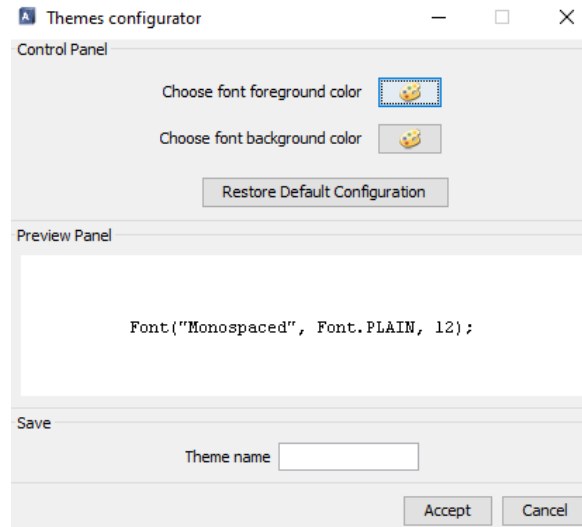


Figura 2-3 Vista para configurar nuevos temas

Una vez hemos accedido a la ventana de configuración, esta se ha programado de manera que podamos cambiar el color de fondo y a su vez el de la fuente de la aplicación.

Cuando entramos en la configuración de colores de la fuente de la aplicación, se nos abrirá la siguiente ventana.

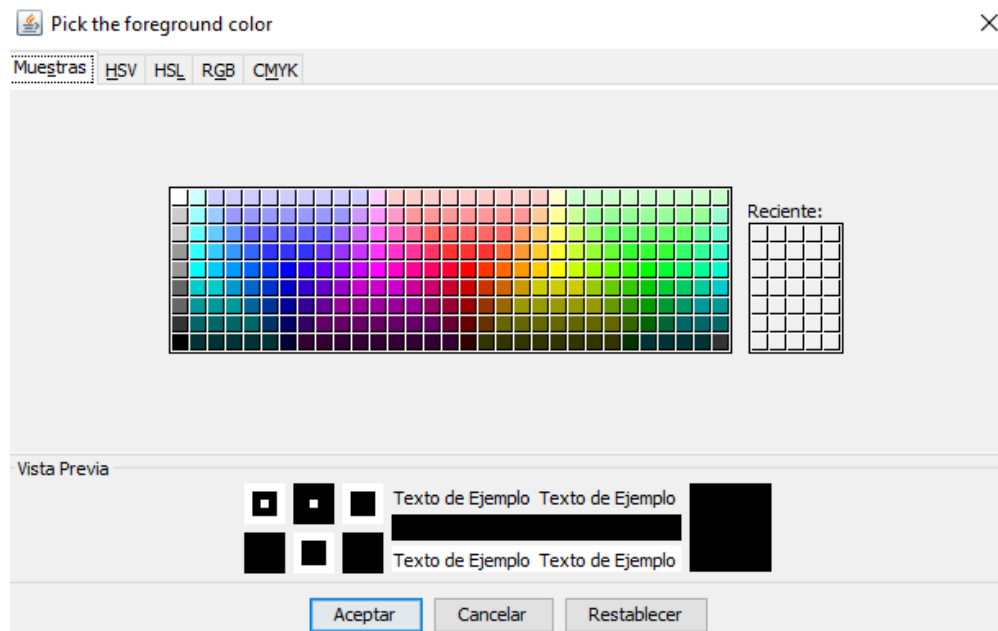


Figura 2-4 Configuración de color de fuente

Una vez dentro, como se puede observar, hay varias opciones a la hora de crear el color que queramos asignarle al texto de ACIDE. En la primera pestaña, se ofrece una gama de colores predefinidos que pueden asignarse. Si por el contrario se prefiere crear un color personalizado, está disponible el resto de las pestañas que aparecen en la captura anterior. En todas ellas es necesario configurar el color en función de la configuración que indican:

- HSV: Matiz, Saturación y Valor
- HSL: Matiz, Saturación y Luminosidad
- RGB: Red, Green and Blue (en español: rojo, verde y azul)
- CMYK: Cyan, Magenta, Yellow y Key

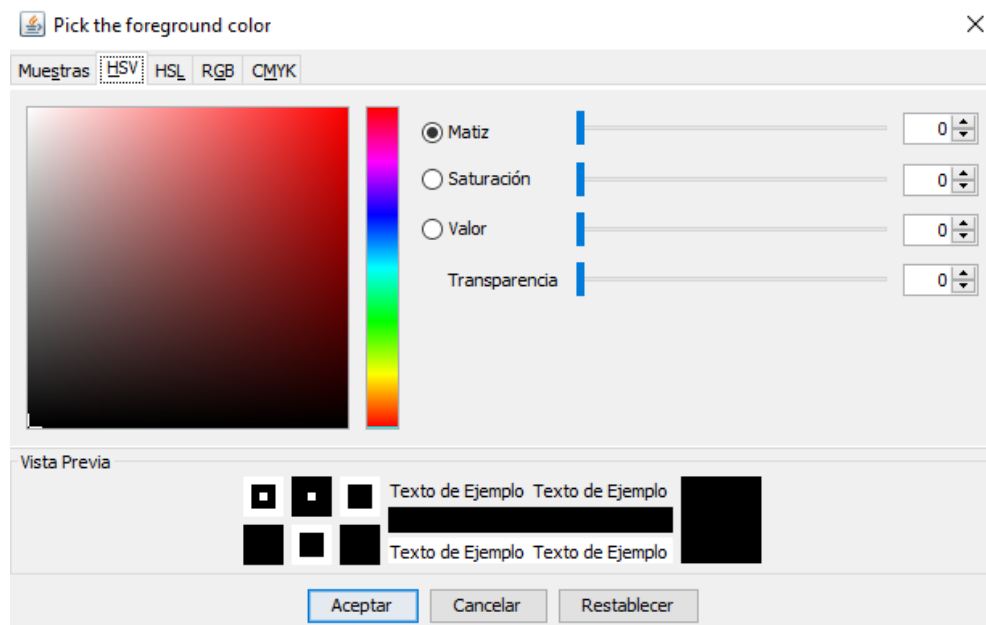


Figura 2-5 Configuración de colores por código HSV

En la captura anterior podemos apreciar el proceso de configuración de un color siguiendo el código HSV para crearlo. Independientemente de la opción que se haya elegido para la configuración de este, una vez tengamos el color idóneo, al darle aceptar volveremos a la ventana anterior (Figura 2-3 Vista para configurar nuevos temas).

Para configurar el color de fondo, hay que realizar el mismo proceso que se ha descrito para el color de la fuente.

Una vez tengamos configurados los colores, se nos mostrará una vista preliminar con la configuración que hemos creado, que nos permitirá valorar, antes de aplicarla a ACIDE, si es necesario modificarla para lograr que todo el contenido se vea de manera óptima.

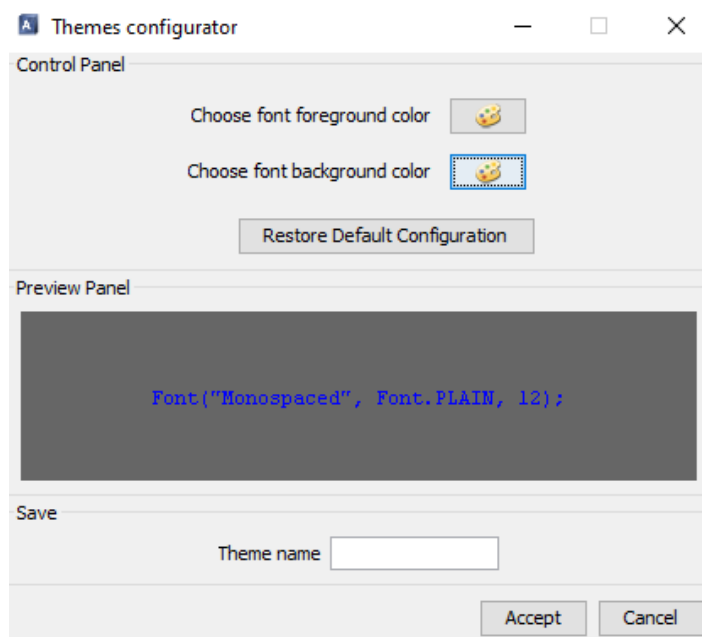


Figura 2-6 Panel vista previa configuración de colores

En la captura anterior se muestra un ejemplo de configuración de un estilo para la aplicación ACIDE. Si la gama de colores que se ha configurado es la deseada, la funcionalidad de gestión de modos de configuración nos plantea dos opciones: mantener el tema en la aplicación de manera persistente, pudiéndolo aplicar en cualquier situación o, simplemente, aplicarlo sin necesidad de guardarlo.

En este punto la aplicación se comporta de la siguiente manera: si el usuario no introduce ningún carácter en el cuadro de texto dedicado al nombre del tema, por defecto, al darle a aceptar, se aplicará la paleta de colores seleccionada a ACIDE, sin guardar la configuración elegida; si, por el contrario, el usuario se decide por rellenar el cuadro de texto, la aplicación emitirá un mensaje solicitando la confirmación para registrar el tema en el sistema de manera persistente.

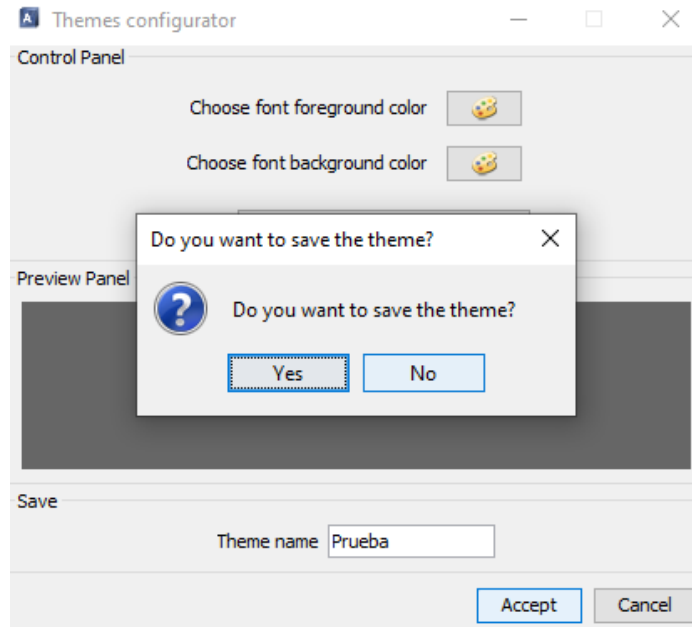


Figura 2-7 Diálogo para confirmar el salvaguardado del tema

Si el usuario finalmente decide guardar el tema, este aparecerá en el menú desplegable dedicado a este apartado, mencionado anteriormente, con el nombre introducido, según podemos observar en la siguiente captura.

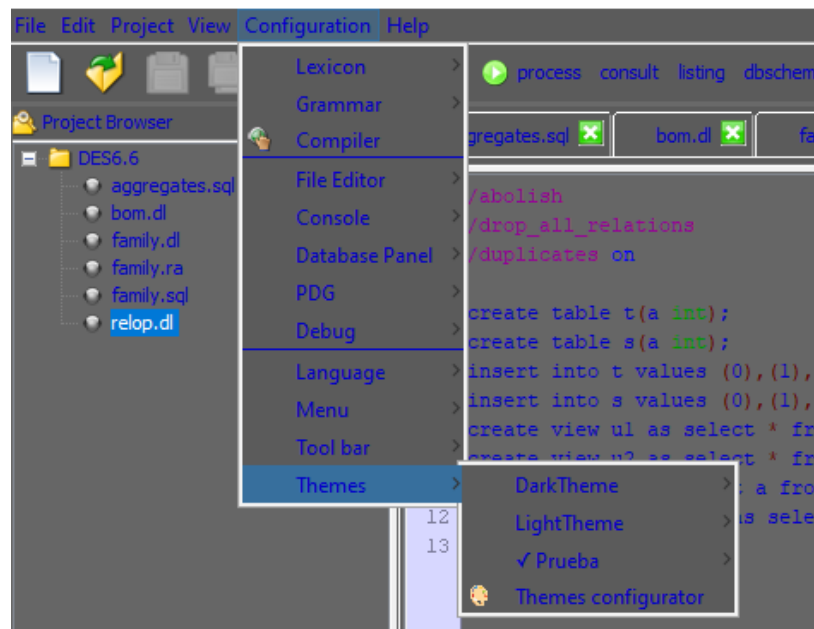


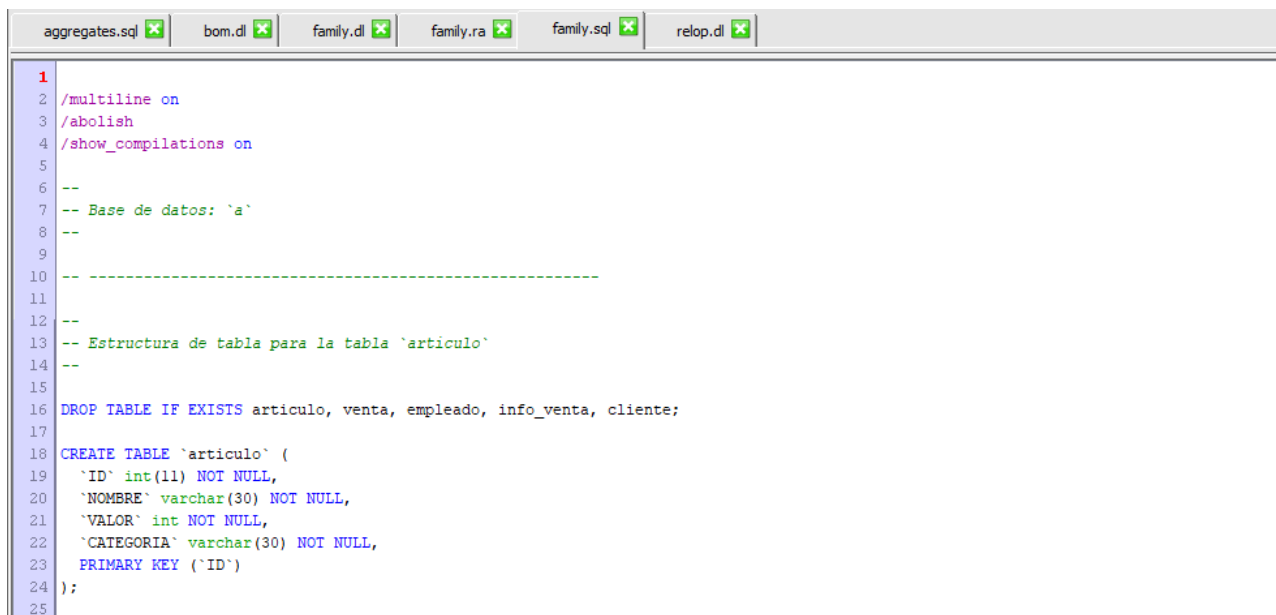
Figura 2-8 Tema registrado en el sistema ACIDE

Además, podemos observar cómo se ha aplicado el tema de manera correcta a todo el entorno.

Como se puede apreciar, el nuevo tema presenta un tick delante del nombre que indica que actualmente es el tema seleccionado. Al cambiar la configuración del diseño de ACIDE, se habilitará la opción de guardar el proyecto, para que, si se quiere, el sistema reconozca ese tema la próxima vez que se inicie y lo aplique automáticamente.

En el momento de crear el tema, se vincula también la configuración de léxico que se quiera, además, si el usuario ha aplicado configuraciones distintas tanto para el editor de archivos, como para la consola, también se añadirán a la configuración del tema. Esto permite que, al configurar temas oscuros, podamos asignar a las palabras clave de cada tipo de archivo una tonalidad que permita diferenciarlas con claridad.

En el siguiente ejemplo se muestra la configuración del léxico para los temas claro y oscuro creados de manera predeterminada en la aplicación.



```
1
2 /multiline on
3 /abolish
4 /show_compilations on
5
6 --
7 -- Base de datos: `a`
8 --
9
10 -----
11
12 --
13 -- Estructura de tabla para la tabla `articulo`
14 --
15
16 DROP TABLE IF EXISTS articulo, venta, empleado, info_venta, cliente;
17
18 CREATE TABLE `articulo` (
19   `ID` int(11) NOT NULL,
20   `NOMBRE` varchar(30) NOT NULL,
21   `VALOR` int NOT NULL,
22   `CATEGORIA` varchar(30) NOT NULL,
23   PRIMARY KEY (`ID`)
24 );
25
```

Figura 2-9 Configuración de léxico tema claro

```

12 --
13 -- Estructura de tabla para la tabla `articulo`
14 --
15
16 DROP TABLE IF EXISTS articulo, venta, empleado, info_venta, cliente;
17
18 CREATE TABLE `articulo` (
19   `ID` int(11) NOT NULL,
20   `NOMBRE` varchar(30) NOT NULL,
21   `VALOR` int NOT NULL,
22   `CATEGORIA` varchar(30) NOT NULL,
23   PRIMARY KEY (`ID`)
24 );

```

Figura 2-10 Configuración de léxico tema oscuro

Por último, significar que la funcionalidad temas nos ofrece opciones para cada uno de los temas almacenados en el sistema.

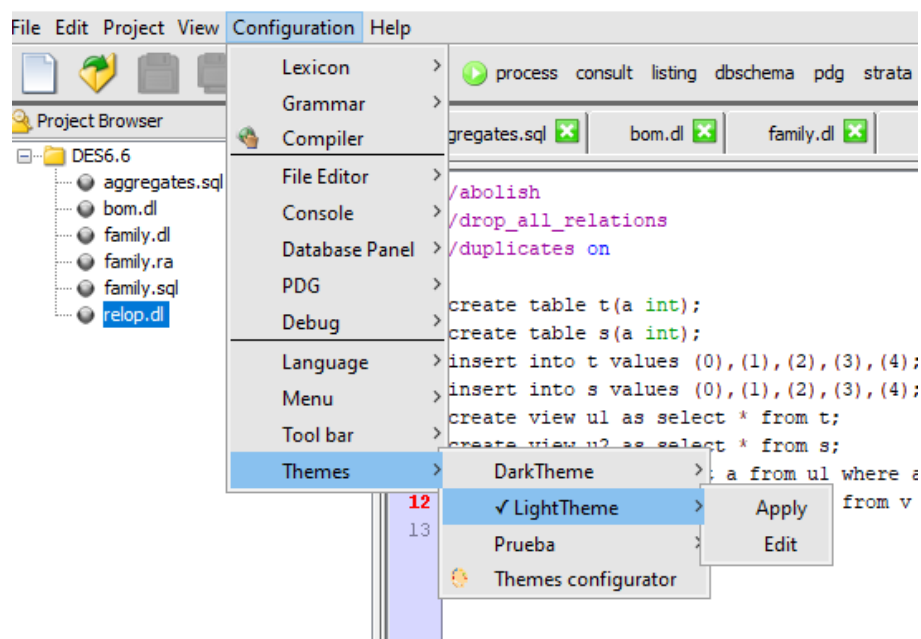


Figura 2-11 Opciones para cada tema

Se nos ofrece la posibilidad de aplicar el tema seleccionado, así como de editarlo.

Si optamos por editar el tema, se nos abrirá la siguiente pantalla.

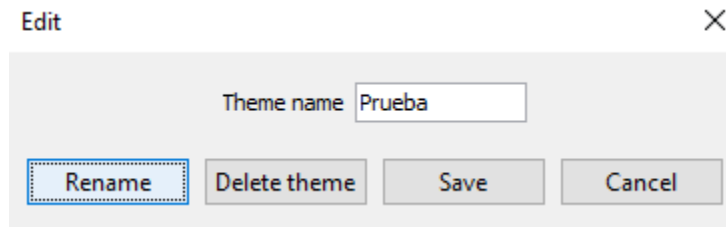


Figura 2-12 Panel edición de tema

Una vez dentro de la vista de edición de un tema, se han implementado las opciones que se muestran arriba. El usuario podrá cambiar el nombre, borrar el tema de manera permanente del sistema, guardar la configuración actual de las vistas para sustituir la configuración inicial de ese tema o, simplemente, cancelar cualquier operación de edición.

Por último, de acuerdo con las exigencias que demandaba la tarea registrada en el documento de tareas pendientes de ACIDE, se ha implementado de tal manera que la configuración de cada uno de los temas guardados en el sistema se almacena en ficheros de texto, que permite su modificación de manera manual, sin necesidad de depender del propio ACIDE. En la siguiente captura se muestra la estructura de un archivo tipo de tema.

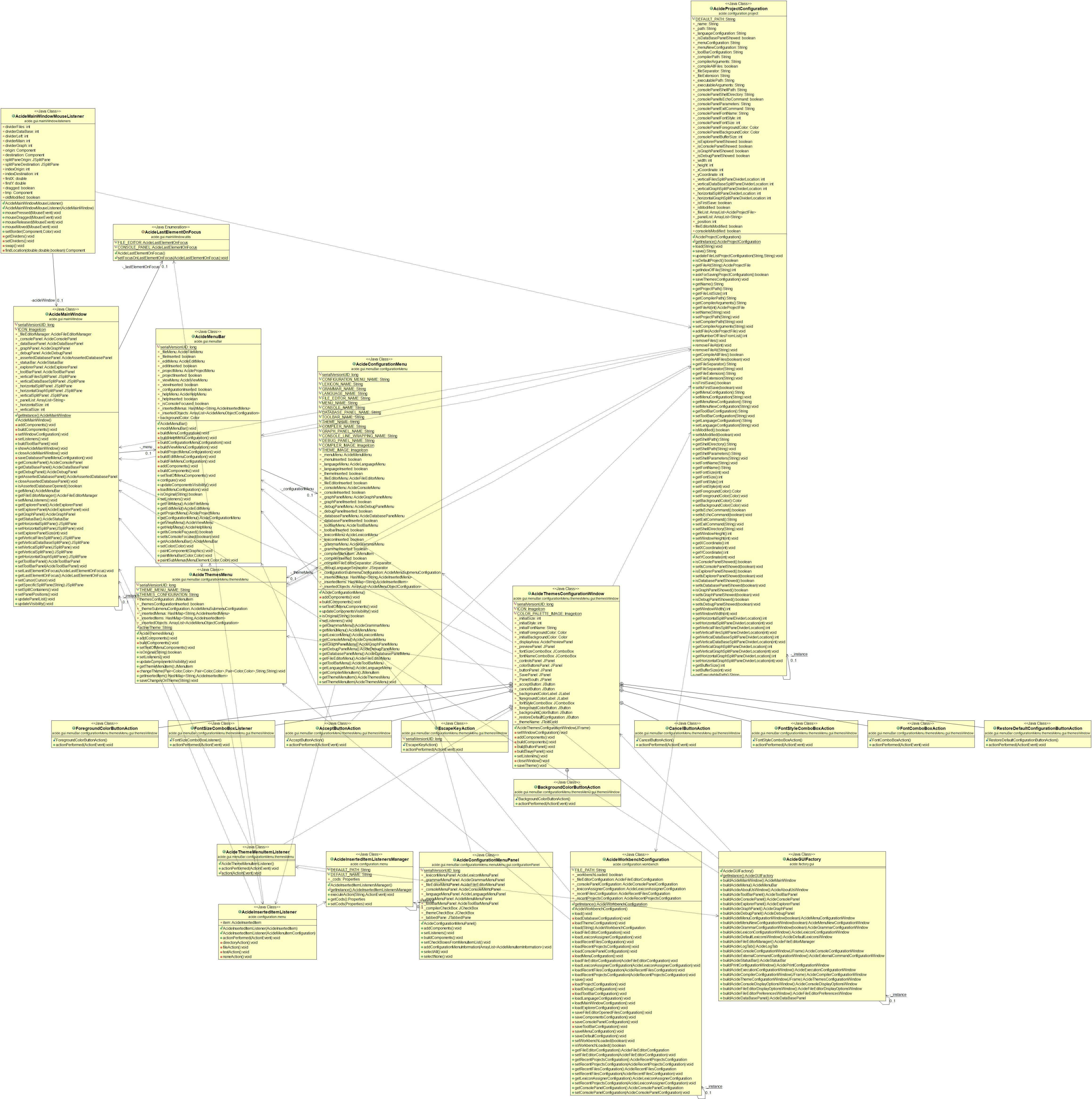
```
#Mon Apr 05 13:46:27 CEST 2021
lexiconFileConfig=./configuration/lexicon/des.xml
foregroundColor=-16777216
backgroundColor=-1
ConsoleforegroundColor=-16777216
ConsolebackgroundColor=-1
lexiconConsoleConfig=./configuration/lexicon/console.xml
FileEditorforegroundColor=-16777216
FileEditorbackgroundColor=-1
```

Figura 2-13 Estructura archivo properties de un tema

- **lexiconFileConfig**: almacena el archivo de configuración léxica aplicada al editor de archivos.
- **foregroundColor**: indica el color de la fuente.
- **backgroundColor**: indica el color de fondo de la aplicación.
- **ConsoleforegroundColor**: indica el color de la fuente de la consola.
- **ConsolebackgroundColor**: indica el color de fondo de la consola.
- **lexiconConsoleConfig**: establece el archivo con la configuración léxica de la consola.
- **FileEditorforegroundColor**: indica el color de la fuente en el editor de texto.
- **FileEditorbackgroundColor**: determina el color de fondo del editor de archivos.

2.1 Diagrama de clases para la funcionalidad principal

A continuación se proceden a explicar las clases necesarias para la implementación de la funcionalidad principal de este trabajo de final de grado, la gestión de temas en ACIDE. Se proporciona el diagrama de clases. Al tratarse de un documento en formato pdf podrá redimensionarse para verse con nitidez.



En primer lugar, se implementó el acceso al menú que ofrece los temas disponibles y la opción para configurar uno nuevo. Para ello, se tuvo que modificar la clase `AcideConfigurationMenu`, donde se alojan todos los ítems del menú de configuración en ACIDE.

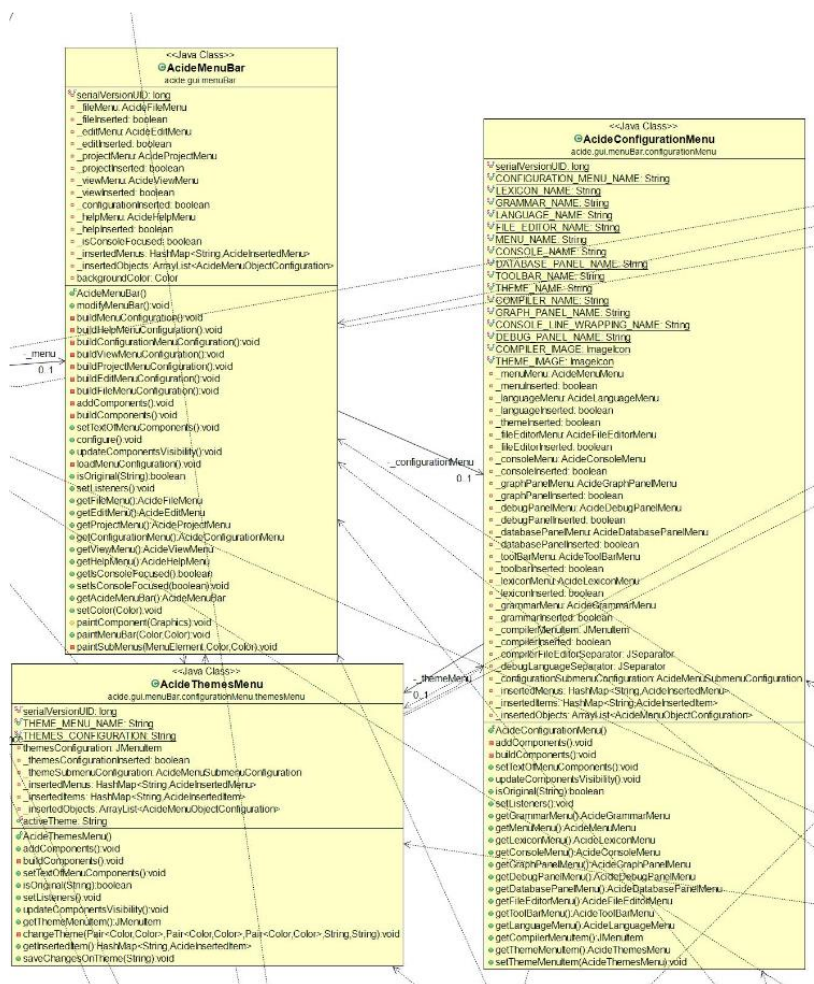


Figura 2-14 Clases afectadas al añadir el menú temas

Se ha creado una relación nueva entre clases las `AcideThemesMenu` y `AcideConfigurationMenu`, fruto de la inclusión en el menú de configuración del nuevo apartado que aloja la gestión de temas. Por otra parte, a pesar de no haber sido representadas en el diagrama, todas las clases de ACIDE que implementan paneles o

menús de la aplicación dependen de AcideThemesMenu, ya que modifica sus atributos a la hora de aplicar los temas.

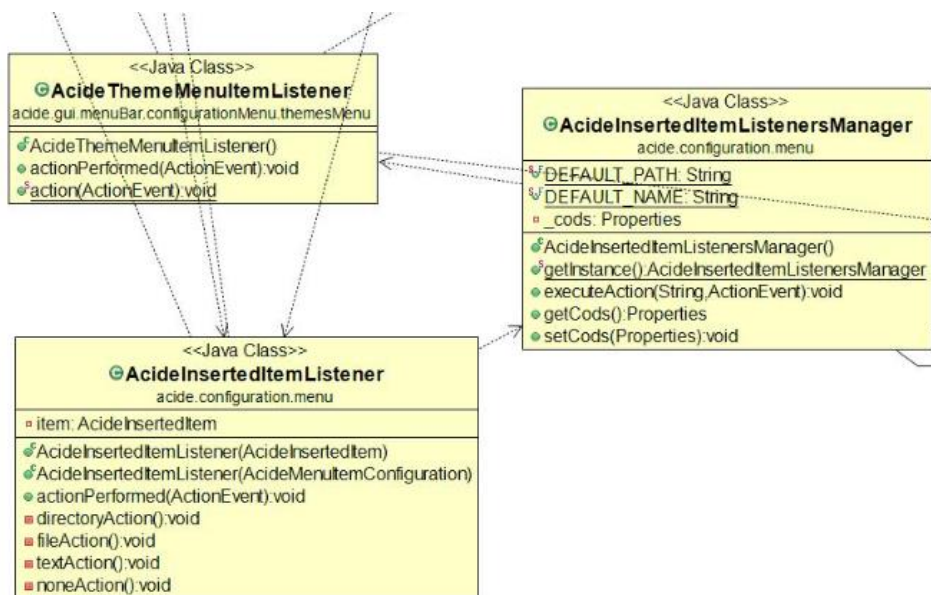


Figura 2-15 Listener que asignan el comportamiento del acceso al menú temas

Para poder acceder a la gestión de temas es necesario configurar los listeners de los elementos del menú, para ello creamos la clase `AcideThemeMenuItemListener`, que extiende de la clase destinada a dar una estructura similar a los listeners de los ítems añadidos.

Por otra parte, se debió desarrollar una ventana que permitiese al usuario configurar los colores referentes a paneles y fuentes de la aplicación. Esta clase queda representada en la siguiente captura bajo el nombre de `AcideThemesConfigurationWindow`.

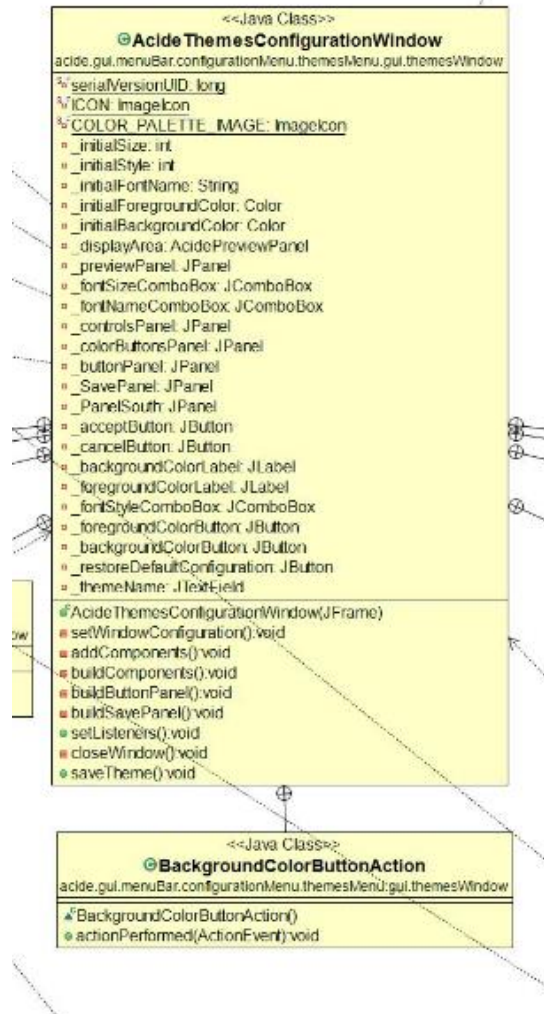


Figura 2-16 Clase de la ventana para la configuración de temas

Por último, podemos apreciar como las clases *AcideWorkbenchConfiguration* y *AcideProjectConfiguration* se relacionan con la clase temas, esto es debido a las opciones de guardado que presentan los temas que se configuran en ACIDE. Se emplean sus métodos para los procesos de carga y guardado del último tema aplicado en la aplicación.

Capítulo 3 - Otras implementaciones y mejoras

Dentro de este capítulo desarrollaré el resto de las características que he añadido a ACIDE, así como los cambios que han contribuido a la mejora de la aplicación.

3.1 Añadir opciones de codificación

Otra de las tareas prioritarias era poder aplicar a los archivos que se abriesen en ACIDE un formato de codificación de caracteres determinado, con el fin de poder representar todos los tipos de caracteres alfanuméricos.

Antes de implementar esta opción, a la hora de cargar ficheros que, por ejemplo, contuviesen letras acentuadas, ACIDE no las reconocía y lo representaba con simbología incorrecta, debido a que no era capaz de determinar qué codificación presentaba el archivo.

Por ello, se ha desarrollado esta funcionalidad, que permite la carga de ficheros reconociendo sus codificaciones y, una vez cargados, se posibilita el cambio de esta.

Ha sido necesario añadir un menú que permite seleccionar con qué codificación representar el fichero que se ha abierto en ACIDE y, al guardarlo, hacer que presente esa codificación concreta. En otras palabras, un archivo que presente una codificación Windows-1252, que no es capaz de representar caracteres acentuados entre otros muchos, podrá cargarse con codificación UTF-8, que sí permite visualizarlos. Posteriormente, al guardar el fichero, quedará con dicha codificación.

Paso a detallar los cambios que se han producido en ACIDE para integrar esta funcionalidad, así como un ejemplo de uso.

Primeramente se valoró la opción de que, al escoger una codificación concreta, esta afectase a todos los archivos abiertos en ese momento en el sistema; sin embargo, tras valorarlo, me pareció más correcto que únicamente afectase al archivo que se estuviese leyendo, con el fin de evitar posibles modificaciones en archivos que no estuviesen a la vista. Es por ello que la opción para cambiar la codificación de un fichero se ha declarado dentro del menú file de ACIDE.

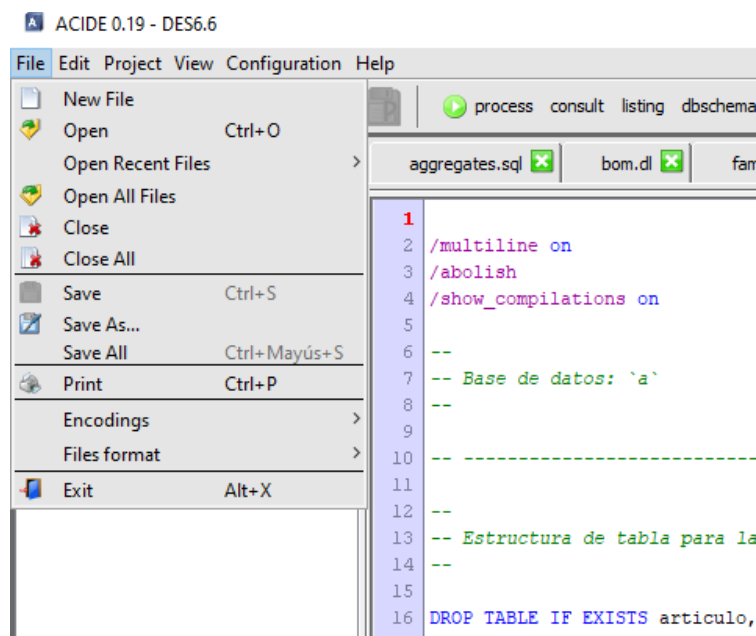


Figura 3-1 Opciones de codificación

Una vez se accede a este menú, las codificaciones que se muestran en la aplicación son las siguientes: UTF-8, UTF-16BE, UTF-16LE, Windows-1252. Aunque se podría añadir cualquiera de las que aparecen registradas en la documentación de JAVA.

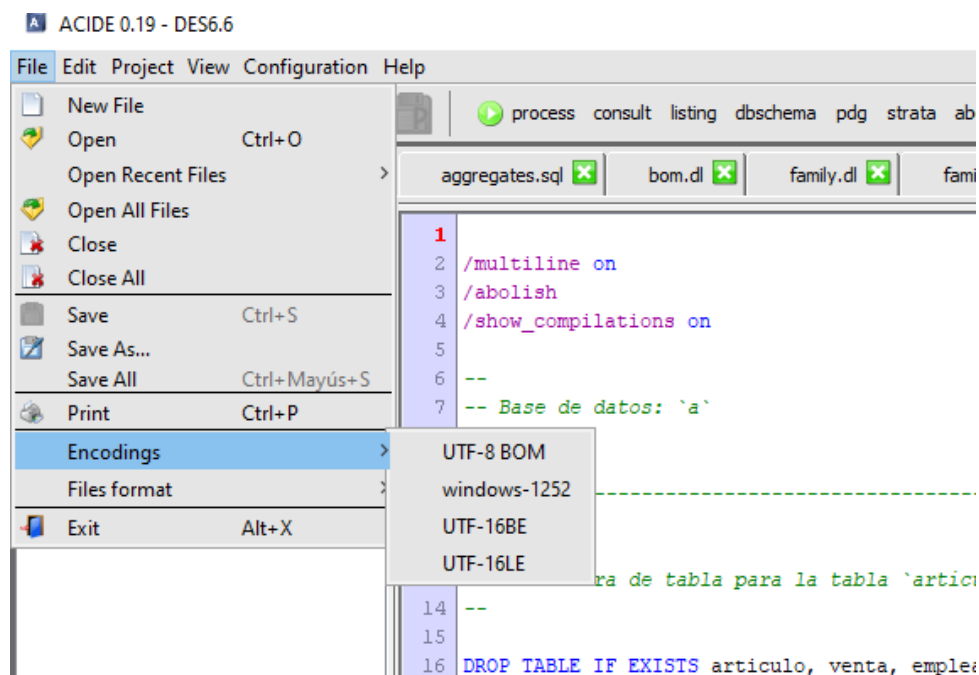


Figura 3-2 Codificaciones disponibles

Al seleccionar la codificación que se quiera aplicar al archivo que esté visible en ese momento, ACIDE se la aplicará, permitiéndonos guardarlo con la configuración de caracteres seleccionada. En las capturas siguientes se muestra un ejemplo de uso.

```
30 INSERT INTO `articulo` ('ID', 'NOMBRE', 'VALOR', 'CATEGORIA') VALUES
31 (1, 'Collar', 200, 'BisuterÃ-a'),
32 (2, 'conchas', 60, 'BisuterÃ-a'),
33 (3, 'falda', 110, 'Ropa'),
34 (4, 'conchas', 60, 'BisuterÃ-a'),
35 (5, 'pulsera', 40, 'BisuterÃ-a'),
36 (6, 'sudadera', 185, 'Ropa'),
37 (7, 'vestido', 205, 'Ropa'),
38 (8, 'camisa', 85, 'Ropa'),
39 (9, 'cartera', 75, 'Cartera'),
40 (10, 'tarjetero', 25, 'Cartera'),
41 (11, 'necesar', 70, 'Cartera'),
42 (12, 'estuche', 95, 'Cartera'),
43 (13, 'bandolera', 125, 'Bolso'),
44 (14, 'Mochila', 300, 'Bolso'),
45 (15, 'Shopper', 150, 'Bolso'),
46 (16, 'riñonera', 70, 'Bolso'),
47 (17, 'maletÃ-n', 180, 'Bolso');
48
```

Figura 3-3 Codificación por defecto windows-1252

```

30 INSERT INTO 'articulo' ('ID', 'NOMBRE', 'VALOR', 'CATEGORIA') VALUES
31 (1, 'Collar', 200, 'Bisutería'),
32 (2, 'conchas', 60, 'Bisutería'),
33 (3, 'falda', 110, 'Ropa'),
34 (4, 'conchas', 60, 'Bisutería'),
35 (5, 'pulsera', 40, 'Bisutería'),
36 (6, 'sudadera', 185, 'Ropa'),
37 (7, 'vestido', 205, 'Ropa'),
38 (8, 'camisa', 85, 'Ropa'),
39 (9, 'cartera', 75, 'Cartera'),
40 (10, 'tarjetero', 25, 'Cartera'),
41 (11, 'necesar', 70, 'Cartera'),
42 (12, 'estuche', 95, 'Cartera'),
43 (13, 'bandolera', 125, 'Bolso'),
44 (14, 'Mochila', 300, 'Bolso'),
45 (15, 'Shopper', 150, 'Bolso'),
46 (16, 'riñonera', 70, 'Bolso'),
47 (17, 'maletín', 180, 'Bolso');
48
49

```

Figura 3-4 Codificación UTF-8 reconocida

Para poder conocer en todo momento cuál es la codificación que está activa en el fichero que se esté visualizando, se ha añadido un campo en la barra inferior de ACIDE, que indica el nombre de esta. En la siguiente captura se muestra un archivo cargado con la codificación UTF-8.

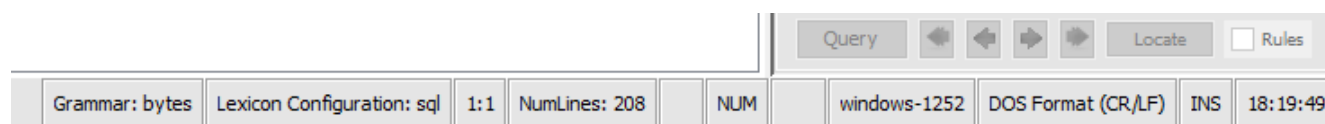


Figura 3-5 Barra tareas con codificación activa

Por último, cabe añadir que ACIDE, al cargar los archivos abiertos en ese proyecto, identificará de manera automática la codificación de los mismos; si no la reconociese, aplicaría por defecto Windows-1252.

Para hacer esto posible se tiene en cuenta la marca de orden de bytes (BOM), un código que representa qué codificación tiene el archivo.

3.2 Permitir mantener las dimensiones de ACIDE

En la versión desde la que partí (v0.18), no era posible que el usuario ajustase las dimensiones de ACIDE y querer hacerlas persistentes a la hora de cerrar el programa, salvo que realizase algún cambio en el proyecto cargado.

Con los cambios que he efectuado, ahora es posible redimensionar todos los paneles disponibles en la vista principal de ACIDE y poder guardarlo; además se ha incluido la posibilidad de configurar el tamaño de la ventana que engloba toda la aplicación y guardar su configuración.

Con esta mejora, se nos permite mantener la relación entre las perspectivas de la aplicación, según las hubiésemos configurado.

3.3 Configurar el botón ejecutar de ACIDE

Dado que ACIDE puede adoptar distintas configuraciones, ha sido necesario implementar la característica que permite ocultar el "botón ejecutar" de la aplicación. Esto hace posible personalizar ACIDE en base a las necesidades que exija la configuración que se vaya a establecer. Dicho botón únicamente debe permanecer visible en las configuraciones de ACIDE con DES [13].

Para el proceso de activar o desactivar este acceso, se ha añadido un checkbox que permite ocultarlo; además, al cargar la aplicación se comprueba si el botón de ejecutar estaba activo, con el fin iniciar el checkbox con el valor indicado.

Para poder acceder a esta nueva opción, será necesario reproducir la ruta Configuración->Barra de herramientas-> Modificar. En la ventana que se nos despliega podremos observar el campo introducido al final de esta, con el estado actual del botón: Marcado, el botón estará oculto; sin marcar, será visible.

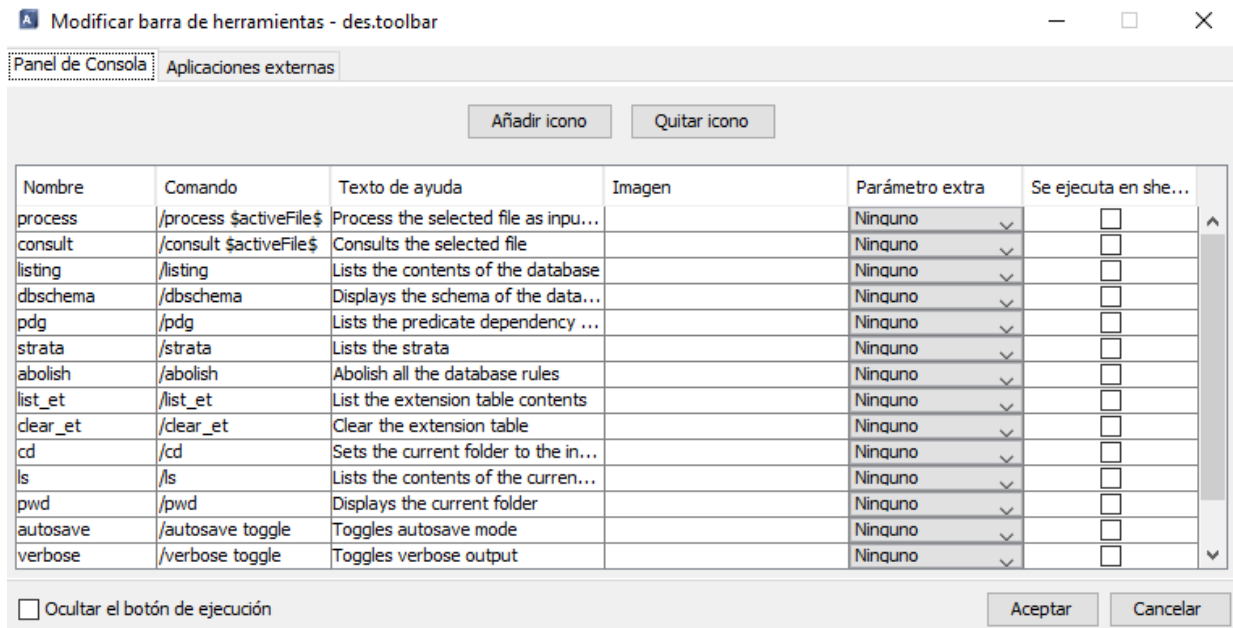


Figura 3-6 Opción para ocultar botón ejecutar ACIDE

A continuación, se muestra un ejemplo de uso de la tarea implementada. Partimos desde una barra de herramientas con el botón visible y pasamos a obtener otra con él oculto.

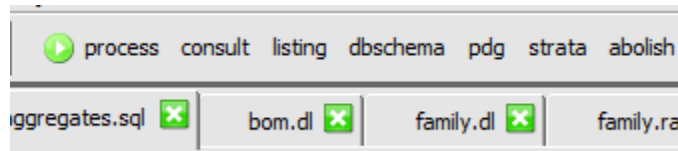


Figura 3-7 Botón ejecutar disponible

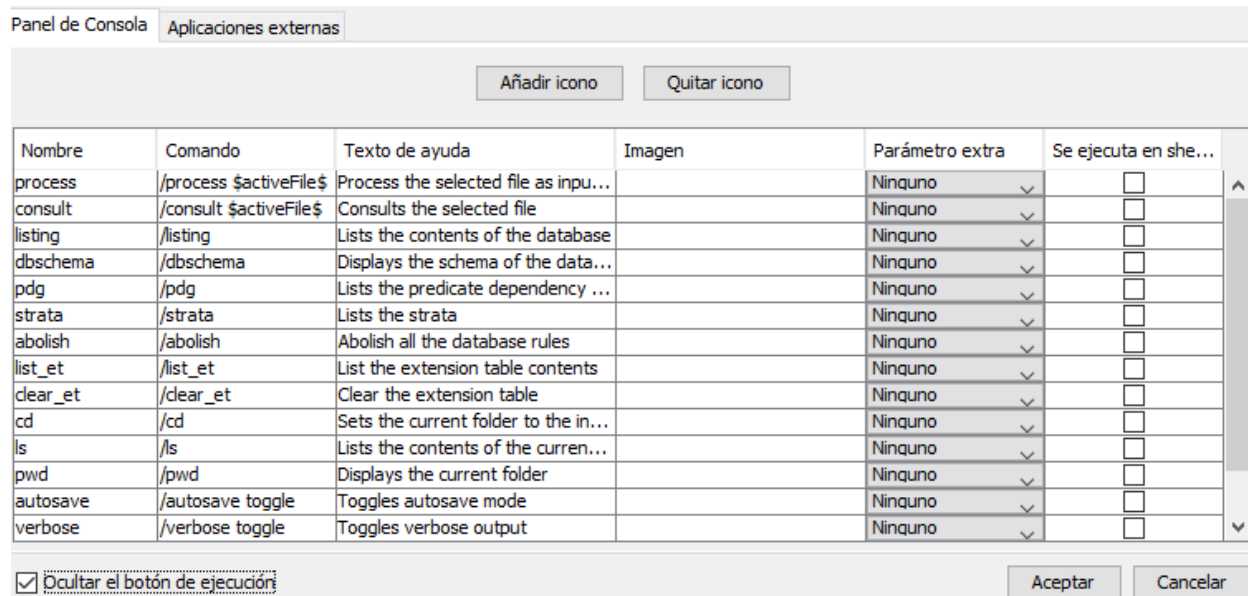


Figura 3-8 Barra de herramientas con botón de ejecutar oculto

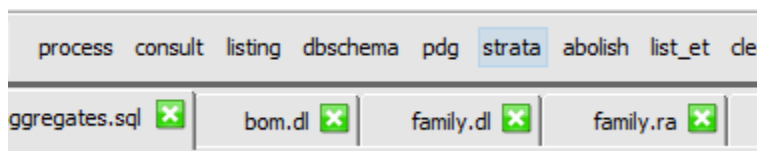


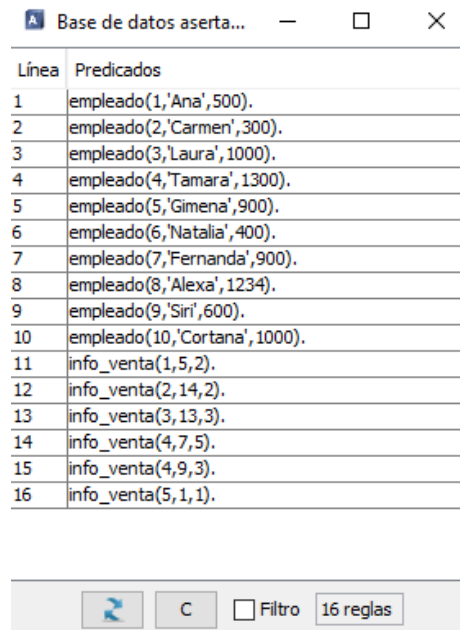
Figura 3-9 Botón de ejecutar oculto

Además, cuando cambiamos la configuración del botón, ACIDE nos preguntará si queremos que se guarde de manera persistente, lo que provocará que, en futuros inicios de la aplicación, este aparezca o esté oculto, en función de la configuración escogida. Si por el contrario no se desea guardar la configuración, únicamente mantendremos esta durante el periodo en el que ACIDE esté abierto.

3.4 Redimensión número de reglas Asserted database

Al abrir la opción asserted database (base de datos asertada) dentro del menú vista, al comienzo del proyecto, el campo que indica el número de reglas, si aparecían demasiadas, cortaba la información, haciendo imposible su consulta.

Se ha procedido a transformar dicho campo para que sea redimensionable, en función del contenido que vaya adquiriendo.



Línea	Predicados
1	empleado(1,'Ana',500).
2	empleado(2,'Carmen',300).
3	empleado(3,'Laura',1000).
4	empleado(4,'Tamara',1300).
5	empleado(5,'Gimena',900).
6	empleado(6,'Natalia',400).
7	empleado(7,'Fernanda',900).
8	empleado(8,'Alexa',1234).
9	empleado(9,'Siri',600).
10	empleado(10,'Cortana',1000).
11	info_venta(1,5,2).
12	info_venta(2,14,2).
13	info_venta(3,13,3).
14	info_venta(4,7,5).
15	info_venta(4,9,3).
16	info_venta(5,1,1).

Figura 3-10 Campo reglas en Base de datos acertada

3.5 Permitir pegar contenido en la consola en la posición del cursor

Cuando el usuario deseaba pegar cualquier texto en la consola, en la posición donde se tuviese el cursor, la aplicación automáticamente insertaba el contenido al final de la línea, sin tener en cuenta la referida posición.

Por ello, se ha corregido dicho comportamiento para que, independientemente de si el usuario decide pegar contenido usando la opción disponible en el menú popup que nos ofrece la consola (Pegar), o a través de la combinación de teclas del teclado Ctrl-V, el texto que se desee incorporar se añada desde la posición en la que el usuario hubiese situado el cursor.

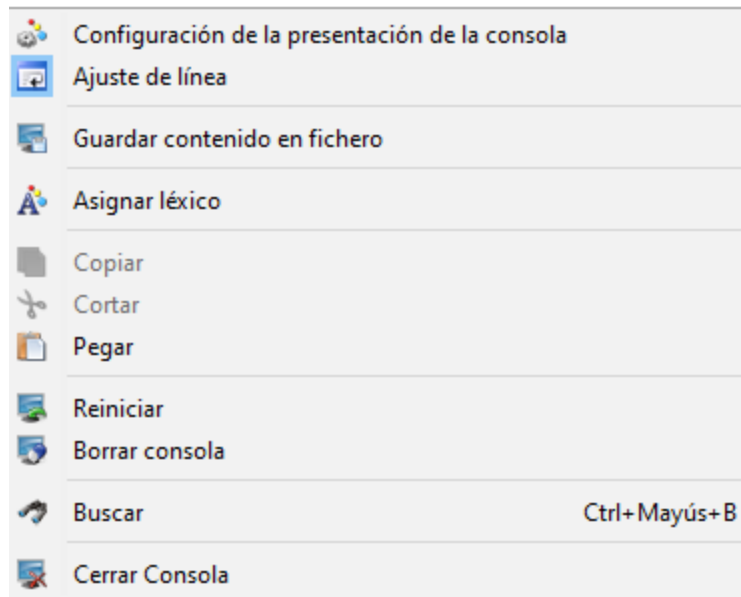


Figura 3-11 Menú popup de la consola. Accesible con el botón derecho del ratón

3.6 Corrección del comportamiento de ACIDE al cerrar un proyecto

El comportamiento esperado en la aplicación al cerrar un proyecto es que se mantenga únicamente la consola definida por dicho proyecto, mientras que el resto de sus archivos y configuraciones dejan de estar disponibles. Sin embargo, al intentar realizar este proceso, también se cerraba la consola, hecho este incompatible con la idea original.

Se ha corregido esta situación para que, al ejecutar esa funcionalidad, ACIDE ahora se comporte de manera correcta, es decir, manteniendo la consola visible.

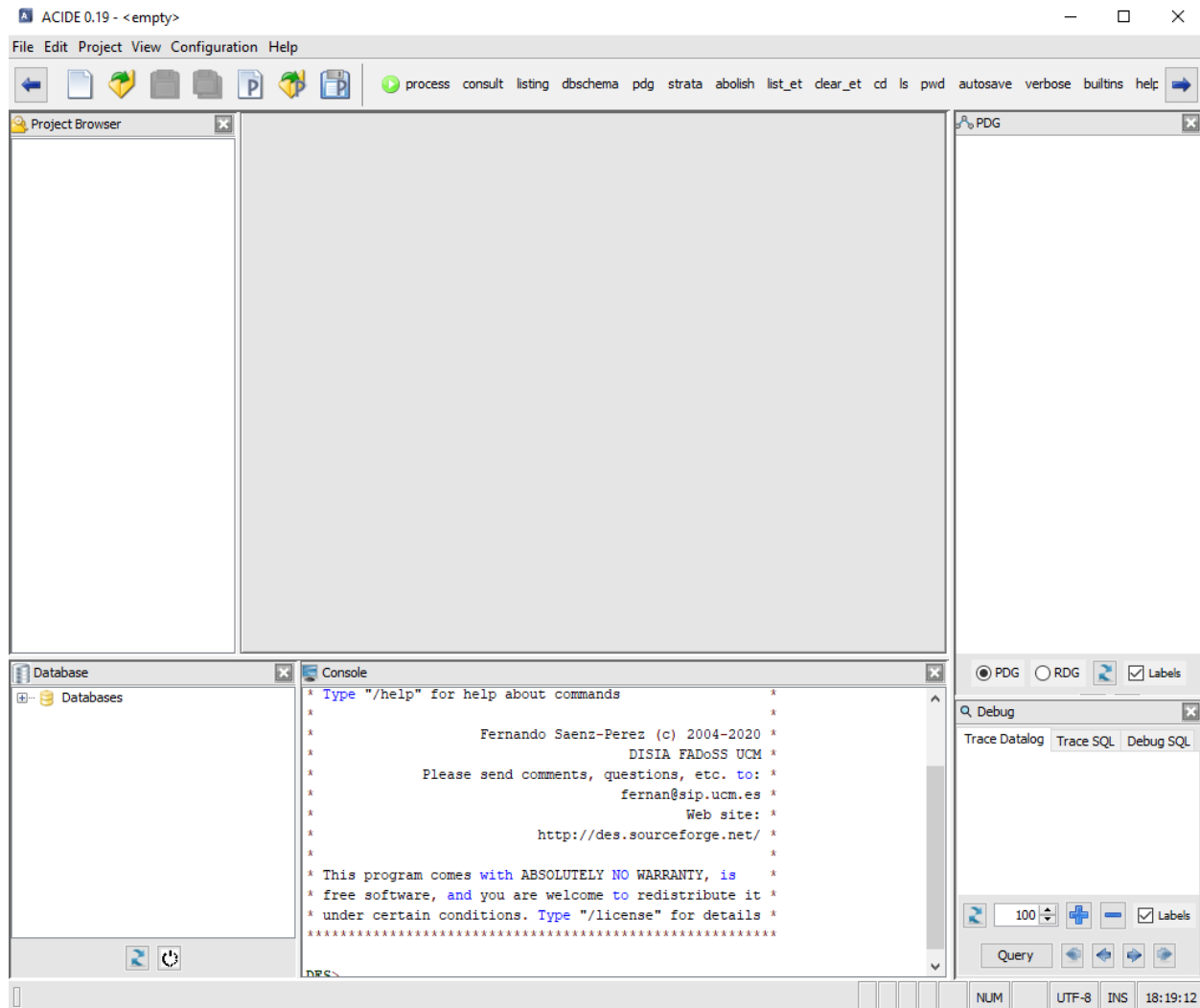


Figura 3-12 Vista ACIDE sin proyecto activo

3.7 Creación de un instalador para ACIDE

Al comenzar con el desarrollo del proyecto era necesario ejecutar ACIDE a través de un archivo .jar, creado en el entorno de desarrollo empleado, en este caso Eclipse [19]. Este proceso dificultaba la labor de prueba, ya que había que tener todas las carpetas con los archivos necesarios para poder iniciar la aplicación. En ocasiones, se originaban fallos derivados de la falta de algún archivo o carpeta con la configuración necesaria para el programa.

Por todo ello, surge la necesidad de poder generar un instalador que permita, además de mantener ACIDE de manera persistente en el sistema, evitar al usuario cualquier error derivado de la falta de archivos esenciales.

Para esta tarea se ha empleado tanto Launch4j [5] para generar el ejecutable en formato exe, como Inno Setup Compiler [4], que permite generar el instalador, ya sea por consola o a través de la interfaz que nos ofrece.

El proceso necesario para generar el instalador se explica en el apéndice 1 de este documento (Creación del instalador con interfaz).

Además, para agilizar el proceso de creación del instalador, se ha creado un archivo bat, configurado de tal manera que, al ejecutarlo, se creará el instalador, abriéndose la pestaña del explorador de archivos en la ruta donde se genera. Este proceso agilizará la futura creación de instaladores para ACIDE.

Al igual que ocurre con el proceso anterior, en el apéndice 2 de este documento (Generación automática del instalador para ACIDE) se explican los pasos a seguir para ejecutarlo, así como el proceso de creación del archivo bat.

```
C:\Windows\system32\cmd.exe
Compressing: C:\CreadorAcide\.\Folders\configuration\toolbar\msdos.toolbarConfig
Compressing: C:\CreadorAcide\.\Folders\configuration\toolbar\newToolBar.toolbarConfig
Compressing: C:\CreadorAcide\.\Folders\configuration\toolbar\sicstusProlog.toolbarConfig
Compressing: C:\CreadorAcide\.\Folders\configuration\toolbar\SWIPProlog.toolbarConfig
Compressing: C:\CreadorAcide\.\Folders\configuration\toolbar\TOY.toolbarConfig
Compressing: C:\CreadorAcide\.\Folders\configuration\toolbar\z3.toolbarConfig
Compressing: C:\CreadorAcide\.\Folders\configuration\workbench\configuration.xml
Compressing: C:\CreadorAcide\.\Folders\examples\a.sql
Compressing: C:\CreadorAcide\.\Folders\examples\aggregates.dl
Compressing: C:\CreadorAcide\.\Folders\examples\aggregates.ra
Compressing: C:\CreadorAcide\.\Folders\examples\aggregates.sql
Compressing: C:\CreadorAcide\.\Folders\examples\bom.dl
Compressing: C:\CreadorAcide\.\Folders\examples\bridge.dl
Compressing: C:\CreadorAcide\.\Folders\examples\bridge1.dl
Compressing: C:\CreadorAcide\.\Folders\examples\b_sin_fechas.sql
Compressing: C:\CreadorAcide\.\Folders\examples\camera.dl
Compressing: C:\CreadorAcide\.\Folders\examples\cm.dl
Compressing: C:\CreadorAcide\.\Folders\examples\db.dl
Compressing: C:\CreadorAcide\.\Folders\examples\ddb.pl
Compressing: C:\CreadorAcide\.\Folders\examples\debug.sql
Compressing: C:\CreadorAcide\.\Folders\examples\e1.sql
Compressing: C:\CreadorAcide\.\Folders\examples\e2.sql
Compressing: C:\CreadorAcide\.\Folders\examples\e3.sql
Compressing: C:\CreadorAcide\.\Folders\examples\e4.sql
Compressing: C:\CreadorAcide\.\Folders\examples\e4trust.sql
Compressing: C:\CreadorAcide\.\Folders\examples\e4trust2.sql
Compressing: C:\CreadorAcide\.\Folders\examples\e5.sql
Compressing: C:\CreadorAcide\.\Folders\examples\empTraining.ra
Compressing: C:\CreadorAcide\.\Folders\examples\empTraining.sql
```

Figura 3-13 Script de creación del instalador

3.8 Crear tabla con Datalog

Tal y como se indicaba en el documento, con todas las tareas pendientes en ACIDE, era imposible crear una tabla aprovechando la funcionalidad que nos ofrecía la aplicación. Según se señalaba en el documento, esta funcionalidad sí estaba disponible en la versión 0.14 del programa.

Se ha subsanado el error, haciendo posible de nuevo crear una tabla con esta herramienta, pudiendo ejecutar las órdenes en el diálogo que se nos muestra al intentar realizar la operación; antes resultaba imposible la escritura de cualquier secuencia en dicho campo.

A continuación, se muestra el error que aparecía y el resultado actual.

```
:-type (
```

Figura 3-14 Error diálogo crear tabla con Datalog

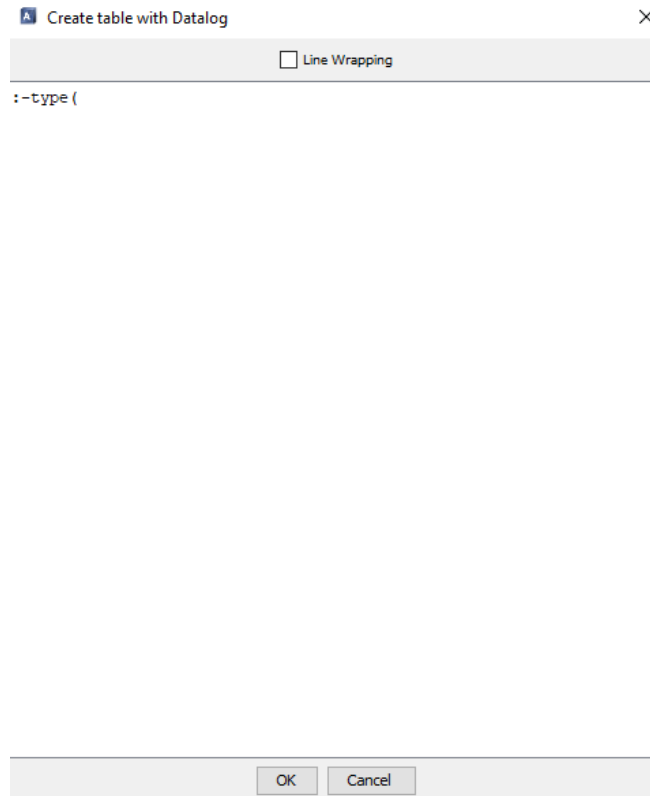


Figura 3-15 Diálogo actual para crear tabla con Datalog

Para poder acceder a esta vista, deberemos situarnos en el panel database, desplegar la base de datos y, con el botón derecho del ratón, desplegar el menú que se nos ofrece, seleccionando a continuación "Crear tabla con Datalog".

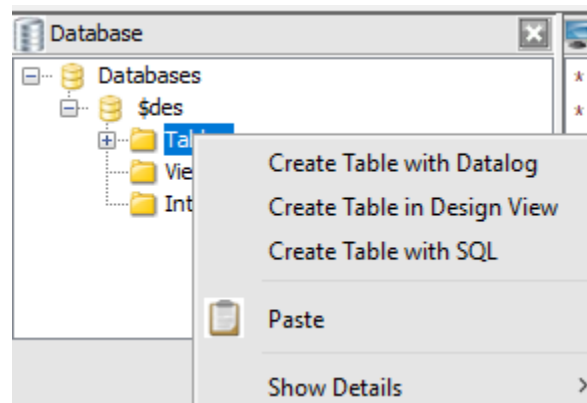


Figura 3-16 Menú desplegable Tablas

3.9 Corrección de la opción de ajuste de línea

En la consola, ACIDE nos ofrece la opción de poder escribir en una línea, sin presentar saltos de línea, o bien, ajustar el tamaño de línea a la vista que se tenga en pantalla en ese momento; con esta opción activa, cada vez que lleguemos al final de la consola, automáticamente se insertará un salto de línea en el campo de texto.

Sin embargo, ACIDE no se comportaba de manera correcta al hacer uso de esta utilidad, al activar o desactivar la función, el foco donde se estuviese escribiendo no se actualizaba y permanecía en el anterior; es decir, si se tenía activada la propiedad y se escribía un texto que ocupase varias líneas, al desactivarla, el cursor permanecía donde estuviese anteriormente, no al final de línea, donde debería estar.

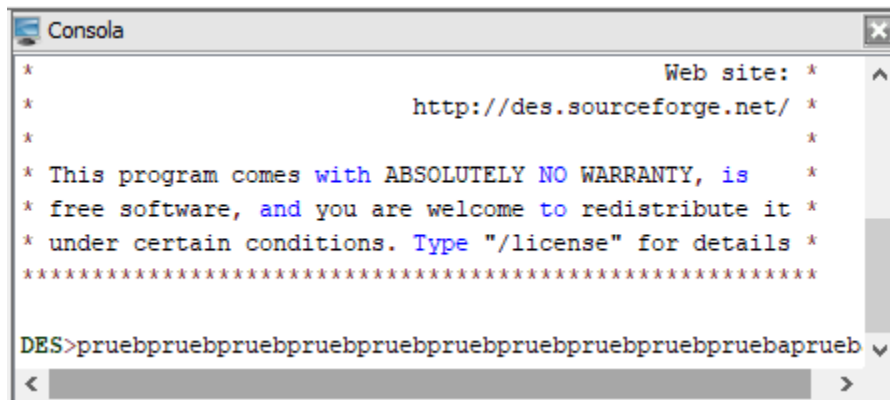


Figura 3-17 Error opción ajuste de línea, no lleva el foco del texto al final

Tras subsanar este fallo, al activar y desactivar la propiedad, ACIDE se comporta de manera adecuada.

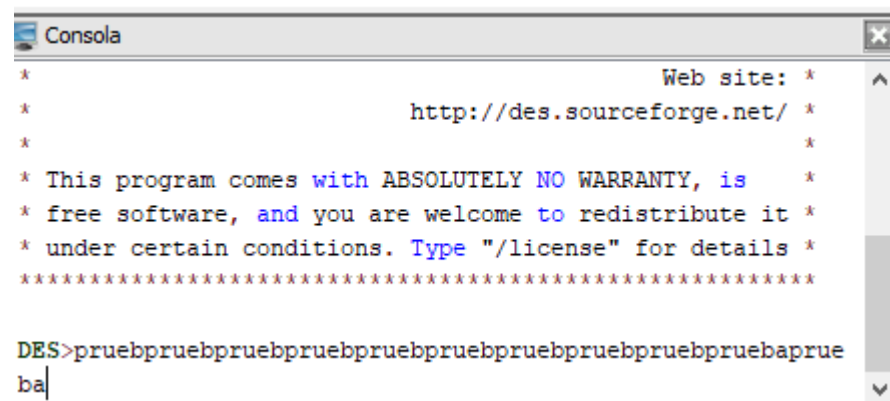
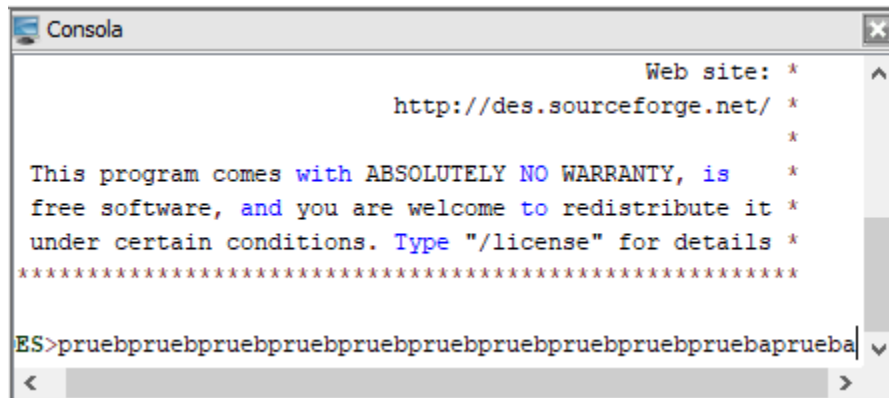


Figura 3-18 Opción de ajuste de línea activa



3.10 Cambio de formato de archivos

Muchos editores nos ofrecen la opción de cambiar el formato de fichero para que sea específico de los sistemas que estemos usando (Windows, Linux o Mac).

Otras de las tareas desarrolladas para ACIDE da la posibilidad de poder seleccionar con qué formato guardar los archivos en el programa. Se han escogido los siguientes tipos: Unix, DOS y Mac.

Para ofrecer esta posibilidad se ha creado una nueva opción en el menú files de ACIDE, que nos permite seleccionar el formato de archivo con el que se guardarán los ficheros. Por defecto, se elige el formato de Windows.

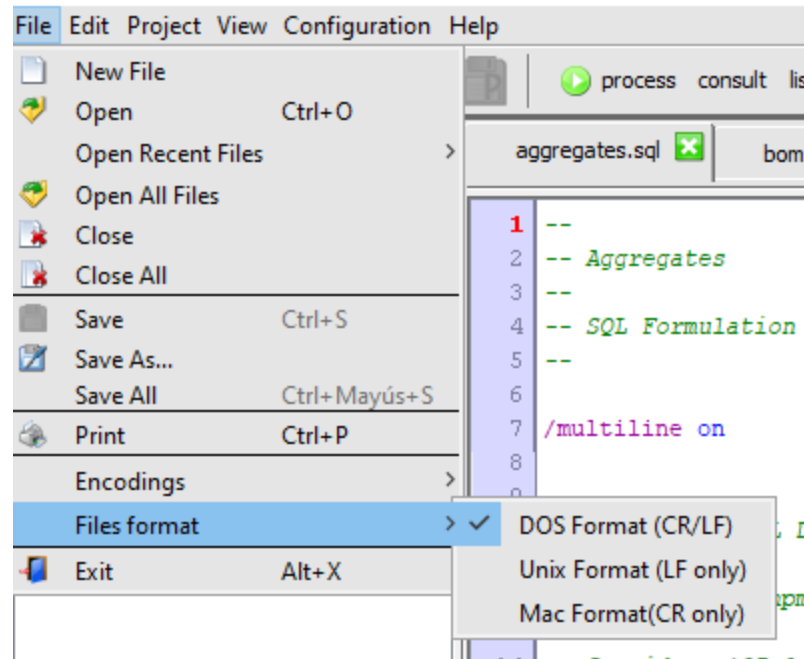


Figura 3-20 Menú formato de archivos

Para poder cambiar los formatos de fichero hay que tener en cuenta los saltos de línea que se producen en su contenido. Con esto claro, aparecen tres tipos de secuencias usadas que definen cada uno de los tres formatos comentados anteriormente.

- **DOS format (CR/LF):** es el formato de fichero de Windows; los saltos de línea vienen representados por la secuencia “\r\n”.
- **Unix format (LF):** Comúnmente usado en sistemas Linux; sus saltos de línea están representados por “\n”.
- **Mac format (CR):** es el formato empleado en los sistemas Mac antiguos; los finales de línea vienen definidos por la secuencia “\r”.

Una vez se decida qué formato emplear y se haya seleccionado, ACIDE aplicará dicho formato al fichero durante el proceso de guardado del archivo.

Hay que señalar que en todo momento podremos saber cuál es el formato activo en la aplicación, para ello se ha añadido a la barra de estado de ACIDE la información correspondiente que nos lo indica.

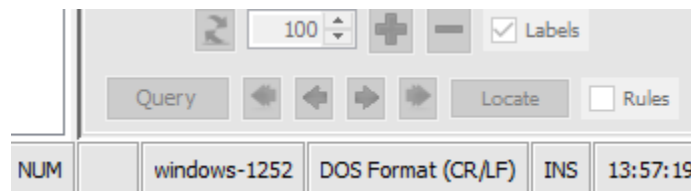


Figura 3-21 Formato DOS seleccionado en ACIDE

3.11 Otras correcciones o mejoras

- Al abrir un proyecto, se implementa el comportamiento que pone el ratón con el cursor de carga hasta que finalice el proceso.
- Se corrige la opción que permite ocultar los menús de ACIDE; no funcionaba de manera correcta. Tras ello se puede elegir qué menús y submenús son visibles en los menús de ACIDE.
- Se añade una opción dentro del menú ayuda de ACIDE, que permite abrir el manual de uso de DES [13].
- Al intentar desplazar las pestañas abiertas en el editor de archivos, se produce un comportamiento inadecuado, apareciendo un cursor de prohibido que impide desplazar la pestaña deseada. Se programa de nuevo esa opción cambiando el código destinado al *actionListener* que detecta los movimientos que se hacen con el ratón en el panel de pestañas. Además, se añade la opción que permite al usuario guardar la nueva disposición de las pestañas.
- Cuando se detecta que un directorio ya tiene un proyecto con el mismo nombre que el que se quiere guardar, al usar la opción "guardar como", se muestra un diálogo en el que se pide confirmación de sobreescritura.

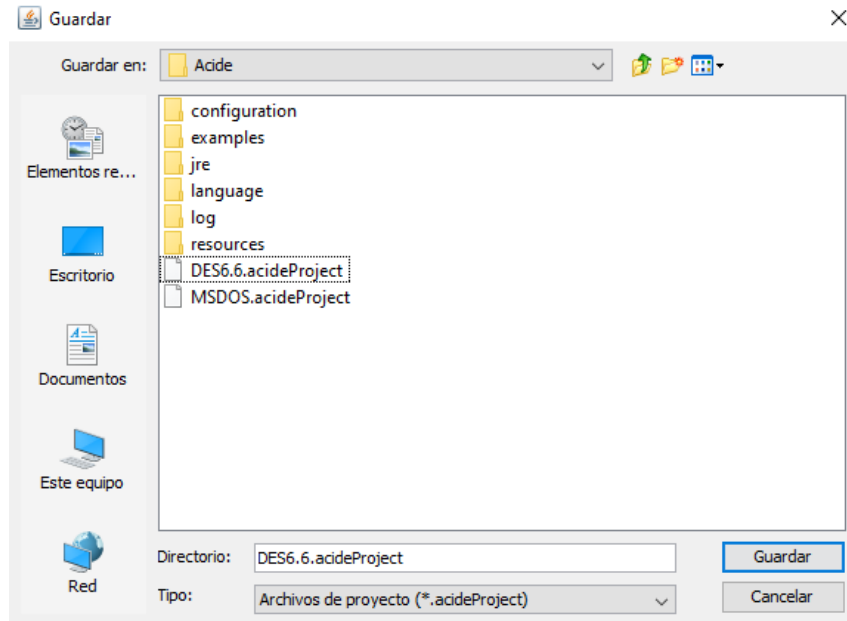


Figura 3-22 Proyecto guardado con el mismo nombre

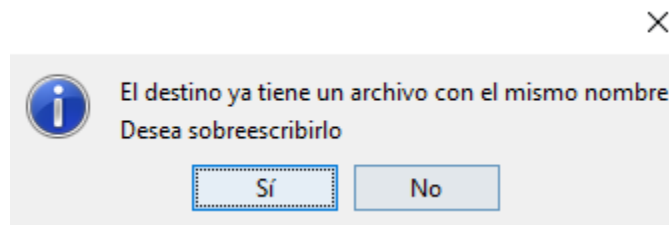


Figura 3-23 Mensaje para confirmar la sobrescritura del proyecto

Capítulo 4 - Conclusiones y trabajo futuro

4.1 Conclusiones

Una vez finalizada mi contribución en ACIDE, y tras repasar los objetivos propuestos al principio del proyecto, considero que han sido desarrolladas todas las tareas marcadas como prioritarias.

Al tratarse de un programa de gran envergadura, en el que han contribuido un gran número de desarrolladores, implementar nuevas funciones conlleva un gran esfuerzo, derivado de la necesidad de adaptación a la estructura y funcionamiento de ACIDE. Además, después de clarificar los conocimientos propios de la aplicación, es necesario mantener los estándares de código acordados en las versiones anteriores.

Todas las tareas añadidas en ACIDE cumplen con los requisitos mencionados anteriormente. A la hora de desarrollar las nuevas características, en particular las que involucran un cambio o adición a la interfaz, se ha empleado el tipo de fuente y la organización adoptados en las versiones anteriores, con el fin de mantener la armonía de ACIDE.

Por último, al dar por terminada mi colaboración, puedo afirmar sin lugar a dudas que ACIDE es un entorno de desarrollo creado para facilitar en gran medida la programación, con un elevado poder de desarrollo, disponiendo de amplias opciones de mejora que lo conviertan en un IDE aún más completo.

4.2 Trabajo futuro

A continuación, se detallan las tareas que en mi opinión dotarían a ACIDE de un mejor comportamiento y mejorarían su experiencia de usuario, así como su funcionamiento:

- **Crear gestión de perspectivas:** que permita personalizar la disposición de los paneles, permitiendo al usuario configurarlos en función, por ejemplo, del tipo de proyecto sobre el que vaya a trabajar.

- **Ordenar la aplicación siguiendo un modelo multicapas:** Estructurar el proyecto siguiendo un modelo multicapa facilitaría en gran medida el futuro desarrollo de ACIDE, así como su mantenibilidad.
- **Disponer de casos de prueba unitarios:** A la hora de probar las tareas implementadas, serviría para comprobar si, en caso de afectar a otras funciones, estas siguen funcionando de manera correcta, simplemente ejecutando la prueba para ese caso de uso.
- **Elaborar diagramas de clases:** Elaborar una documentación que aporte diagramas facilitaría la interpretación y el entendimiento del código sobre el que se vaya a trabajar.
- **Disponer de la opción de abrir ficheros haciendo drag&drop:** Sería interesante permitir al usuario poder abrir archivos sin necesidad de navegar hasta la ruta donde esté disponible, simplemente arrastrarlo sobre el editor de archivos y que ACIDE lo abra.

Todas estas propuestas son las que considero más importantes desde mi punto de vista para la mejora sustancial de ACIDE, no pensando únicamente en funcionalidad, sino también en el proceso de desarrollo. No obstante, hay multitud de mejoras disponibles en el documento TODO_ACIDE que se aporta al inicio del proyecto, para que contemplemos cuáles son las principales necesidades que se demandan cada año.

Chapter - Introduction

This report describes the implementation of the themes management in ACIDE as the main task of the final degree project. In addition, it will describe the rest of the functionalities that have been developed, contributing to an improvement in the functioning of the application, as well as in the user experience.

In the respective sections, each of the implemented functionalities will be explained in detail.

At the end of the document there is an explanatory guide to the process of generating the application installer.

5.1 Motivation

During the years that I have been studying software engineering, I have always had a clear vocation for the development of applications and software, which has been accentuated over time.

ACIDE was the first option I considered, as it is a project that fully responds to my concerns.

The purpose of this work is to provide ACIDE with features that give it a better user experience, as well as a better functioning. As they are not focused on a specific topic, the functionalities developed have been framed under priority tasks pending completion.

The lack of systems that allow the debugging of SQL databases favours the development of ACIDE. DES [13] allows us to carry out textual debugging, which, together with the graphic interface tool offered by ACIDE for programming and debugging databases, makes it necessary to make continuous progress in the process of improving the application, consolidating itself as a tool of great help for possible programmers. In order to increase the usability of the application and to continue creating a user-friendly interface, the process of improving and adding new functionalities to ACIDE has continued.

5.2 Objectives

The objectives of this final degree project are the implementation of several functionalities pending development and the correction or improvement of some of the existing ones.

As the main functionality, a theme management system has been developed that allows ACIDE to be configured with the desired range of colours for the backgrounds and text of panels and menus. In addition, this configuration can be saved as a theme of the application and can be applied at any time. How this works will be described in more detail in later sections.

Many other features have also been implemented and will be mentioned later. To name a few: the possibility of displaying the files uploaded to the application under the encodings it offers; the resizing of the application's panels and windows and the possibility of making it persistent when exiting the application; allowing the hiding of the application's submenus from the ACIDE configuration menu; linking the lexicon to each theme created; and adding the DES [13] help option that opens a file with the manual.

5.2.1 Objectives not achieved

In the development of this project it has not been possible to fix the error that occurs when starting ACIDE: the console does not start correctly. The expected behaviour should be as shown in the following screenshot (Figura 1-1 Comportamiento correcto de la consola).

When starting ACIDE, randomly, without a constant pattern, the `$success` command appears, and the console must be restarted to show the result of the previous capture. (Figura 1-2 Error consola al iniciar ACIDE)

During the time spent trying to fix this bug, the error was identified as coming from the order of execution of the threads that ACIDE creates on start-up. However, it has not been possible to delimit which invocation of the commands executed by each of them causes this output.

5.3 Background of the project

ACIDE is an integrated development environment that can be easily used as an interpreter, compiler or database system for Relational Algebra, TRC, DRC, Datalog and SQL.

This project is a continuation in the development of the tool, which has been developed by previous groups:

- The first year in which it was carried out was in the 2006-2007 academic year by Diego Cardiel Freire, Juan José Ortiz Sánchez and Delfín Rupérez Cañas.
- During the 2007-2008 academic year the development was continued by Miguel Martín Lázaro.
- The next iteration in the development of ACIDE was in the academic year 2010-2011 by Javier Salcedo Gómez.
- During the academic year 2012-2013 the development of ACIDE continued in the hands of Pablo Gutiérrez García-Pardo, Elena Tejeiro Pérez de Ágreda and Andrés Vicente del Cura.
- In the academic year 2013-2014 the development was carried out by Juan Jesús Marqués Ortiz, Fernando Ordás Lorente and Semíramis Gutiérrez Quintana.
- During the 2014-2015 academic year the development was carried out by Sergio Domínguez Fuentes.
- And during the 2019-2020 academic year the development was carried out by Sergio García Rodríguez.

This project has always been directed by Fernando Sáenz Pérez.

5.3.1 Development of ACIDE

In order to assess the magnitude of ACIDE, the evolution of the project from its beginnings to the current version is detailed below.

ACIDE was born from the need to create a configurable IDE for different programming languages, offering the user a graphical tool with a multitude of intelligible

options. As stated in the first reports dedicated to the development of ACIDE, some text editors, such as Crimson Editor [14] and JEdit [15], were taken as a reference.

Continuing with the ACIDE approach, it was identified what kind of development environment should be created to fulfil the initial idea of the application. As it is an IDE designed for different programming languages, environments not specialised in a single language, in this case Eclipse [19], were taken as a reference. However, being an application that limits its tools to the plugins offered by its developers, and which is sometimes tedious to use, other applications focused on specific languages that could offer interesting features were evaluated in order to facilitate the use of ACIDE as much as possible. This group includes software such as: JBuilder [16], JCreator [17] and C++ Builder [18].

In the next evolution of ACIDE, Visual Studio Shell [20] was taken as a reference, which makes it possible to adapt the application to its own language and to create customised tools.

Later, WinEdt [21] and NetBeans [22] applications were added as references for the development of ACIDE, in addition to the previous ones.

During the subsequent development of ACIDE, the need arose to link the console with ODBC and DES [13]; for this purpose, the graphic engines of the database management systems MS Access [23], Oracle [24] and Postgres [25] were taken as a reference. In addition, ASPIDE [26], with similar functionalities to ACIDE, began to be taken into account.

The evolution continued with the implementation of various functionalities taking advantage of the existing link with DES [13] and the ODBC system. New panels were created to provide ACIDE with more options, as well as enhancement tasks to correct or improve existing ones. Many of the features offered by ACIDE at the time were based on the aforementioned Crimson Editor [14].

Next, there was a desire to increase the functionality of the project by adding French translation and creating a new panel for future debugging. Work was also carried

out to improve the options already available in order to simplify their use as much as possible and to provide ACIDE with a user-friendly interface.

Finally, the version from which I started had an extension of the debugging present in the last version of DES [13] available, which allowed configuring the state of tables and views that are not selected, improving the usability of the system, as well as maintenance tasks and bug fixes in ACIDE.

5.3.1.1 Source code standards

As this is a project under development by multiple developers, it has been necessary to implement code standards. Its purpose is to be able to implement new functionalities or improvements under the same programming scheme, which greatly facilitates the future addition of new options or the maintainability of ACIDE.

Firstly, all comments and attributes have been declared in English. In addition, at the beginning of the classes present in the project, the licence, GPLv3, is indicated. (Figura 1-3 Código de licencia ACIDE)

Secondly, comments have been added to each of the functions and attributes that have been added to the code, in order to indicate their purpose and functionality.

In addition, each of the Java classes in the project has a Javadoc code in the following format: (Figura 1-4 Código Javadoc de las clases).

Thirdly, all the variables declared in the classes in ACIDE begin with "_", in order to homogenise the nomenclature of the attributes that are declared. In addition, the name of the classes starts with Acide, followed by a descriptive name of its functionality. All the words that make up the name will begin with a capital letter, thus respecting the Java standard. (Figura 1-5 Estándar para el nombre de las clases)

The names of declared methods adopt the same format as described for classes.

Fourthly, when it is necessary to declare constant variables in the code, their name must be capitalised; if they are represented by several words, they are separated using "_". (Figura 1-6 Formato para las variables constantes)

And in fifth and last place, all classes that refer to application windows must necessarily include the following methods: Figura 1-7 Métodos obligatorios para nuevas ventanas

5.4 Work plan

At the beginning of the project, the director, Fernando Sáenz Pérez, provided me with a file with the latest version of ACIDE on which to start working, as well as a document with the pending tasks to be carried out, ordered by the highest priorities.

Based on this, I began to develop the functions or features following the order of priority stipulated in the document. Every week I communicated to the director the status of the project and the progress made.

Once I had finished the functionalities, I generated an executable of the application to send it to the director for further validation. In this way, I have been improving and extending the features of ACIDE until the current version of the system was achieved (ACIDE 0.19).

As an additional comment, it should be noted that, after developing several improvements or functions, another intermediate version (ACIDE 0.18.1) has been published. This version of the application is available on the ACIDE website (<http://www.fdi.ucm.es/profesor/fernan/ACIDE/>). Here you can find the versions of ACIDE configured for each of the operating systems (Windows, Linux and Mac), a user manual, a file with the changes implemented in the latest version and various sections of the website that offer different options, such as: screenshots of the application, information about the developers who have contributed, statistics of ACIDE downloads and pre-built bundles. (Figura 1-8 Web de ACIDE)

Chapter - Conclusions and future work

6.1 Conclusions

After having completed my contribution to ACIDE, by compiling the objectives proposed at the beginning of the project, all the tasks marked as priorities have been developed.

As this is a large-scale program, to which many developers have contributed, implementing new functions entails a great effort, derived from the need to adapt to the structure and operation of ACIDE. In addition, once the application's own knowledge is clear, it is necessary to maintain the code standards agreed in previous versions.

All the tasks added in ACIDE comply with the requirements mentioned above. When developing all new features, particularly those involving a change or addition to the interface, the font type and organization adopted in previous versions has been used in order to maintain the harmony of ACIDE.

Finally, in concluding my contribution, I can say that ACIDE is a development environment created to greatly facilitate programming, with a high development power, with extensive enhancement options to make it an even more complete IDE.

6.2 Future work

Here are the tasks that, in my opinion, would make ACIDE behave better and improve the user experience, as well as its operation:

- **Create perspective management:** that allows customizing the layout of the panels, allowing the user to configure them according to, for example, the type of project he/she is going to work on.

- **Order the application according to a multi-tier model:** Structuring the project following a multi-tier model would greatly facilitate the future development of ACIDE, as well as its maintainability.

- **Having unit test cases:** When testing the implemented tasks, it would be useful to check if, in the event of affecting other functions, these continue to work correctly, simply by executing the test for that use case.

- **Elaborate class diagrams:** Creating documentation that provides diagrams would facilitate the interpretation and understanding of the code to be worked on.

- **Having the option to open files by drag&drop:** it would be interesting to allow the user to open files without having to navigate to the path where it is available, simply drag it onto the file editor and ACIDE will open it.

All these improvements are the ones that I consider most important from my point of view for the substantial improvement of ACIDE, not only in terms of functionality, but also in the development process. However, there are a multitude of improvements available in the TODO_ACIDE document, which is provided at the beginning of the project so that we can contemplate which are the main needs that are demanded each year.

BIBLIOGRAFÍA

- [1] [*Datalog Educational System V6.6 User's Manual. Fernando Sáenz Pérez*](#)
- [2] [Java encodings charset possibilities](#)
- [3] [Google Drive](#)
- [4] [Inno Setup Compiler](#)
- [5] [Launch4j](#)
- [6] Sergio García Rodríguez. Debugging SQL. Universidad Complutense de Madrid. 2020
- [7] Sergio Domínguez Fuentes. ACIDE Debugging. Universidad Complutense de Madrid. 2015
- [8] D. Cardiel Freire, J. J. Ortiz Sánchez y D. Rupérez Cañas. ACIDE: A Configurable IDE. Universidad Complutense de Madrid. 2007
- [9] M. Martín Lázaro. ACIDE 0.2: a configurable integrated development environment. Universidad Complutense de Madrid. 2008
- [10] J. Salcedo Gómez. ACIDE: A Configurable IDE. DES GUI Front-end. Universidad Complutense de Madrid. 2011
- [11] P. Gutiérrez García-Pardo, E. Tejeiro Pérez de Ágreda, A. Vicente del Cura: A Configurable IDE. DES GUI Front-end. Universidad Complutense de Madrid. 2013
- [12] J. J. Marqués Ortiz, F. Ordás Lorente y S. Gutiérrez Quintana: A Configurable IDE. DES GUI Front-end. Universidad Complutense de Madrid. 2014
- [13] Página oficial de DES: <http://www.fdi.ucm.es/profesor/fernan/des/index.html>
- [14] Página oficial de Crimson Editor: <http://www.crimsoneditor.com>

- [15] Página oficial de JEdit: <http://www.jedit.org>
- [16] Página oficial de JBuilder: <http://www.borland.com/jbuilder>
- [17] Página oficial de JCreator: <http://www.jcreator.com>
- [18] Página oficial de C++ Builder: <http://www.borland.com/cppbuilder>
- [19] Página oficial de Eclipse: <http://www.eclipse.org/>
- [20] Página oficial de Visual Studio Shell: <http://msdn.microsoft.com/en-us/library/vstudio/bb685691.aspx>
- [21] Página oficial de WinEdt: <http://www.winedt.com/>
- [22] Página oficial de NetBeans: <https://netbeans.apache.org/>
- [23] Página oficial de MS Access: <http://office.microsoft.com/es-es/access/>
- [24] Página oficial de Oracle: <http://www.oracle.com>
- [25] Página oficial de Postgres: <https://www.postgresql.org/>
- [26] Febbraro, O., Reale, K., Ricca, F.: ASPIDE: Integrated Development Environment for Answer Set Programming. In: Delgrande, J., Faber, W. (eds.) LPNMR 2011, LNAI 6645, pp. 317-330, 2011. Página oficial de ASPIDE: <http://www.mat.unical.it/ricca/aspide/>

APÉNDICES

Apéndice A - Creación del instalador con interfaz

7.1 Introducción

La finalidad de este documento es explicar de manera detallada los pasos a seguir para la generación de un instalador, que permitirá incorporar de manera persistente ACIDE a nuestro sistema.

7.2 Requisitos previos

Antes de comenzar a desarrollar el instalador, será necesario disponer de los dos programas empleados para su creación: Inno Setup Compiler [4] y Launch4j [5].

A continuación, se explica la utilidad de cada una de las dos aplicaciones anteriormente citadas.

- **Inno Setup Compiler:** Esta aplicación se ha usado para la creación del propio instalador, sin embargo, para poder desarrollarlo será necesario que tengamos preparados los archivos y carpetas necesarias para el correcto funcionamiento de nuestro programa, así como un ejecutable .exe capaz de iniciarlo.
- **Launch4j:** La finalidad principal de este programa es convertir nuestro archivo .jar en un ejecutable .exe.

Además de los dos requisitos anteriores se necesitará disponer de la versión JRE de java para ejecutar nuestra aplicación, en este caso se dispone de la 1.8.

Enlaces para la descarga de todos los puntos mencionados:

JRE:

https://javadl.oracle.com/webapps/download/AutoDL?BundleId=244554_d7fc238d0c_bf4b0dac67be84580cfb4b

Inno Setup Compiler: <https://jrsoftware.org/isdl.php#stable>

Launch4j: <https://sourceforge.net/projects/launch4j/files/latest/download>

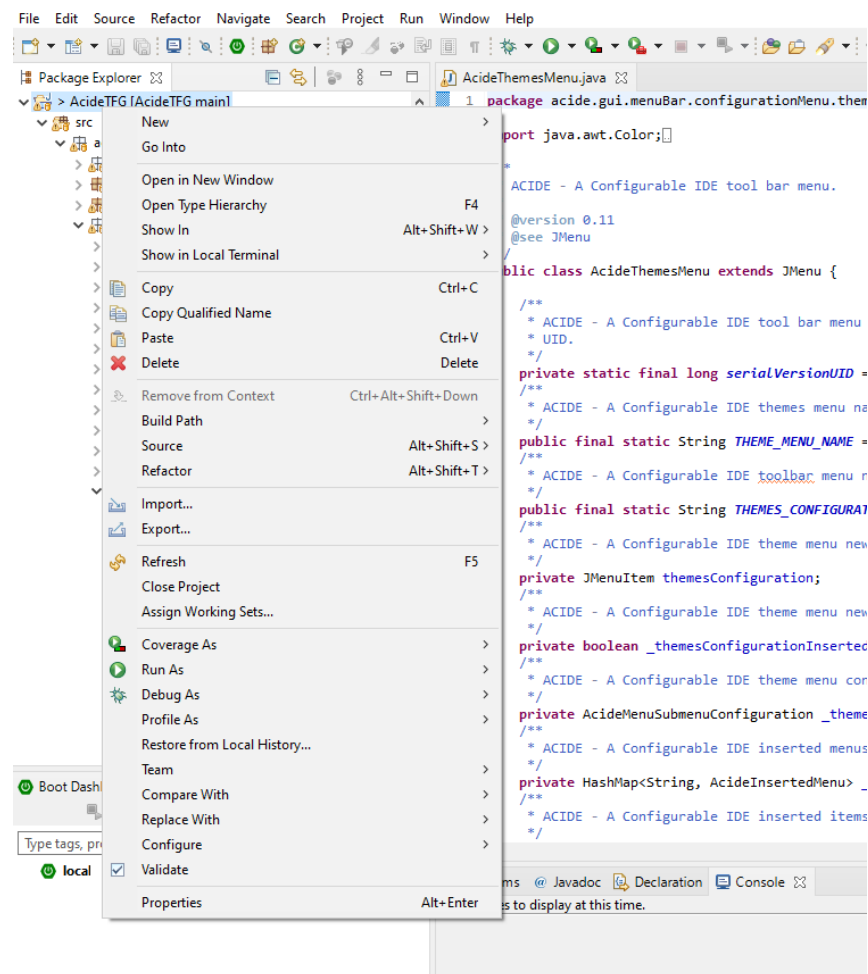
7.3 Proceso de creación

A continuación, se detallan los pasos necesarios durante el proceso de creación del instalador.

7.3.1 Generación de archivo jar

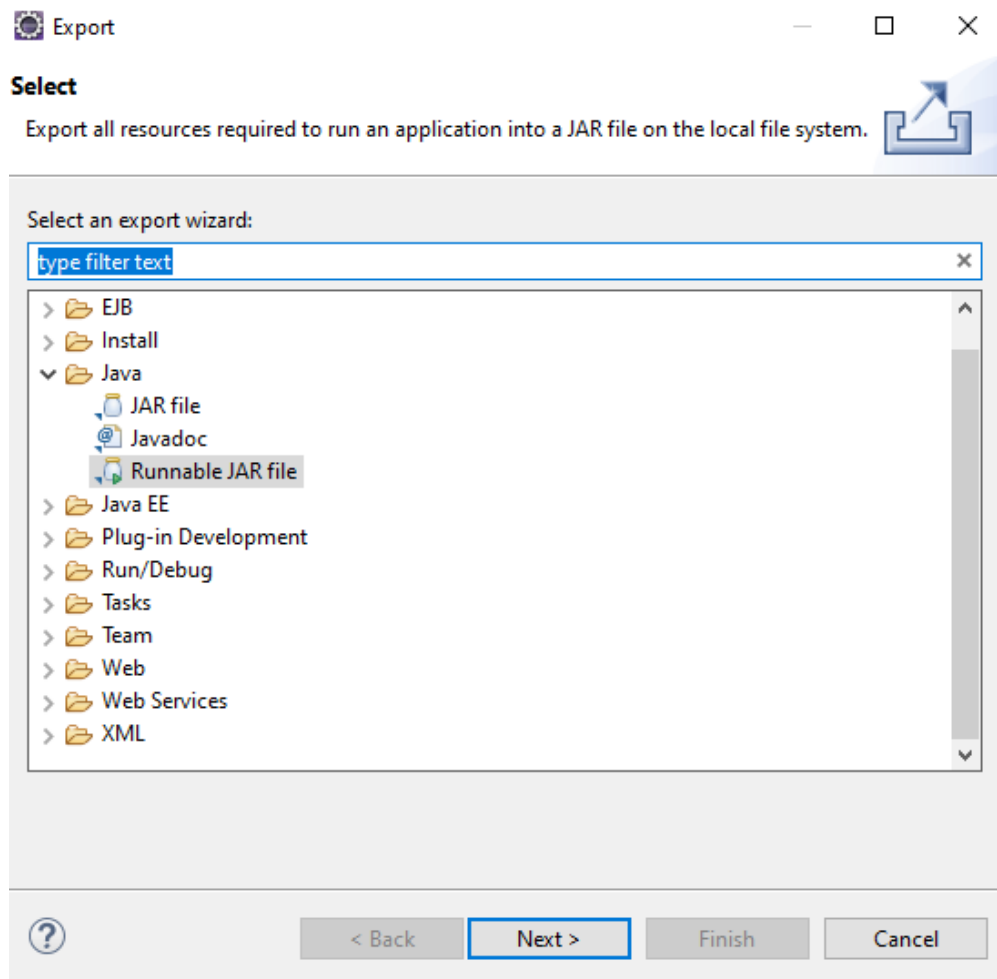
Al tratarse de una aplicación Java, será necesario generar el archivo ejecutable en formato jar. Para ello, usaremos Eclipse [19], aprovechando que es la herramienta empleada para codificar la aplicación.

Una vez abierto Eclipse [19], pulsaremos con el botón derecho del ratón sobre la carpeta que contiene nuestro proyecto.



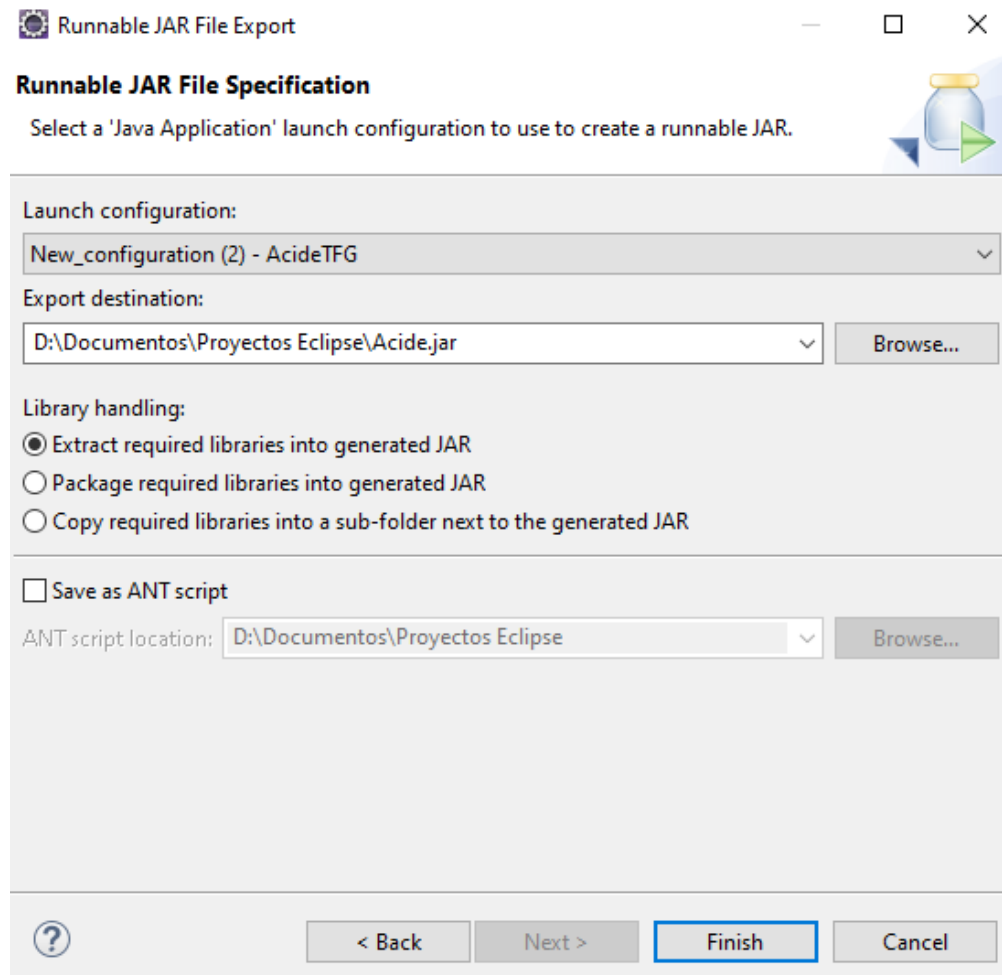
1 Menú desplegable del proyecto

En el menú desplegable que se nos abre, accederemos con el botón izquierdo del ratón a la opción export.



2 Menú export del proyecto

En la ventana de export nos desplazaremos hasta el desplegable Java y, una vez dentro, escogeremos la opción Runnable JAR file.



3 Menú opción export Runnable JAR file

En de la ventana que nos aparece al escoger la opción anterior, deberemos elegir la configuración de lanzamiento del programa (Launch configuration) y la ruta donde queremos que se nos guarde el archivo jar que generaremos. El resto de las opciones que nos aparecen las dejaremos por defecto. Por último, haremos click en finish.

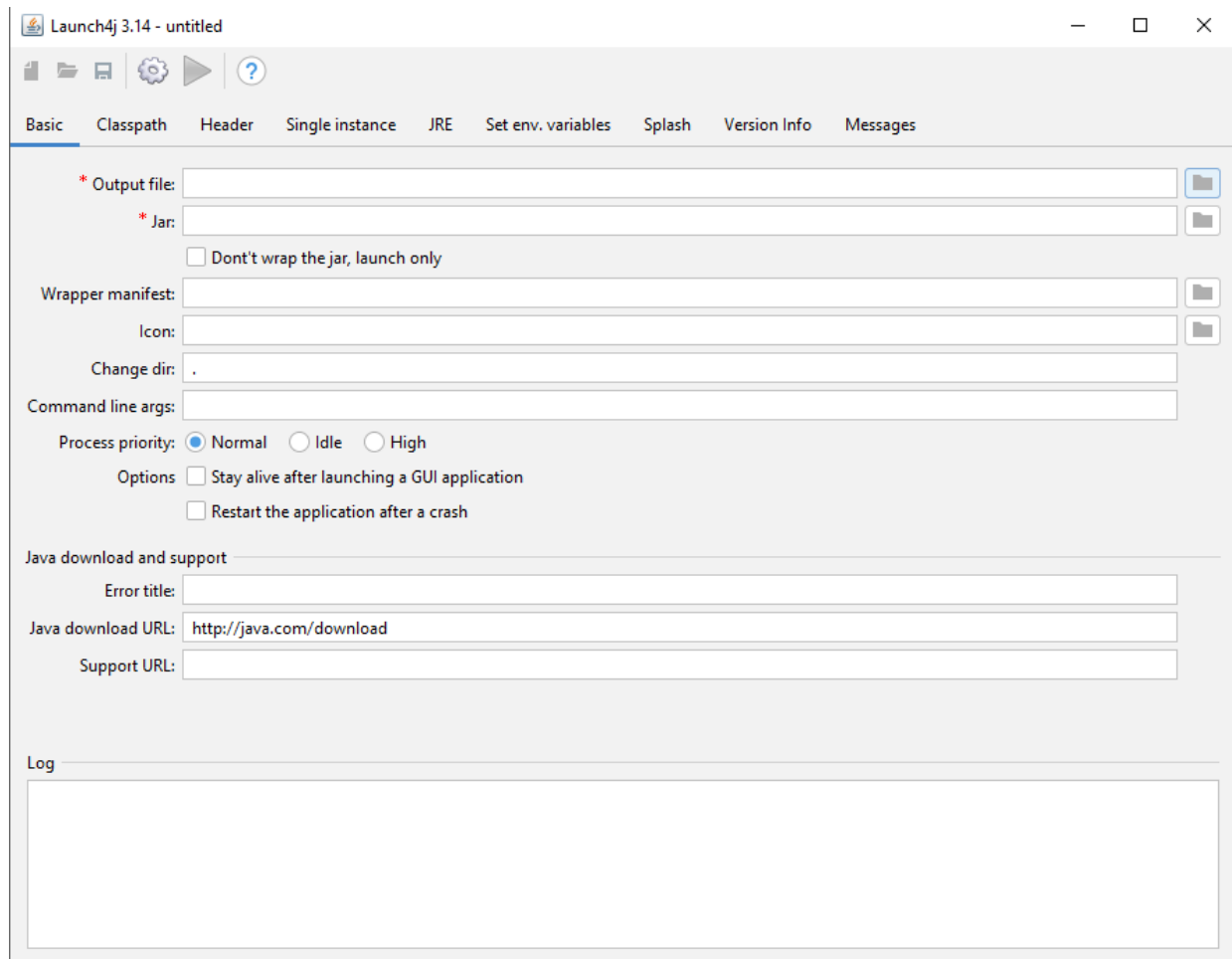
Seguidamente, comenzará el proceso para crear el archivo jar. Cabe la posibilidad de que aparezcan ventanas que nos informen de eventuales fallos durante el desarrollo; lo obviaremos y daremos OK a todos estos mensajes.

Finalmente, deberíamos tener nuestro ejecutable jar correctamente creado en la ruta indicada en el paso anterior.

7.3.2 Generación del ejecutable .exe:

Una vez tengamos el jar creado, será necesario disponer de un ejecutable .exe para más adelante poder vincularlo con nuestro instalador. Para este proceso usaremos el programa Launch4j [5].

Al abrir esta aplicación nos aparecerá la siguiente ventana:



4 Vista principal Launch4j

Dentro de esta pestaña deberemos preocuparnos por configurar de manera correcta las opciones marcadas con los asteriscos en color rojo (Output file y Jar).

La primera opción, Output file, sirve para escoger la ruta donde guardar nuestro archivo ejecutable y el nombre con el que nos aparecerá.

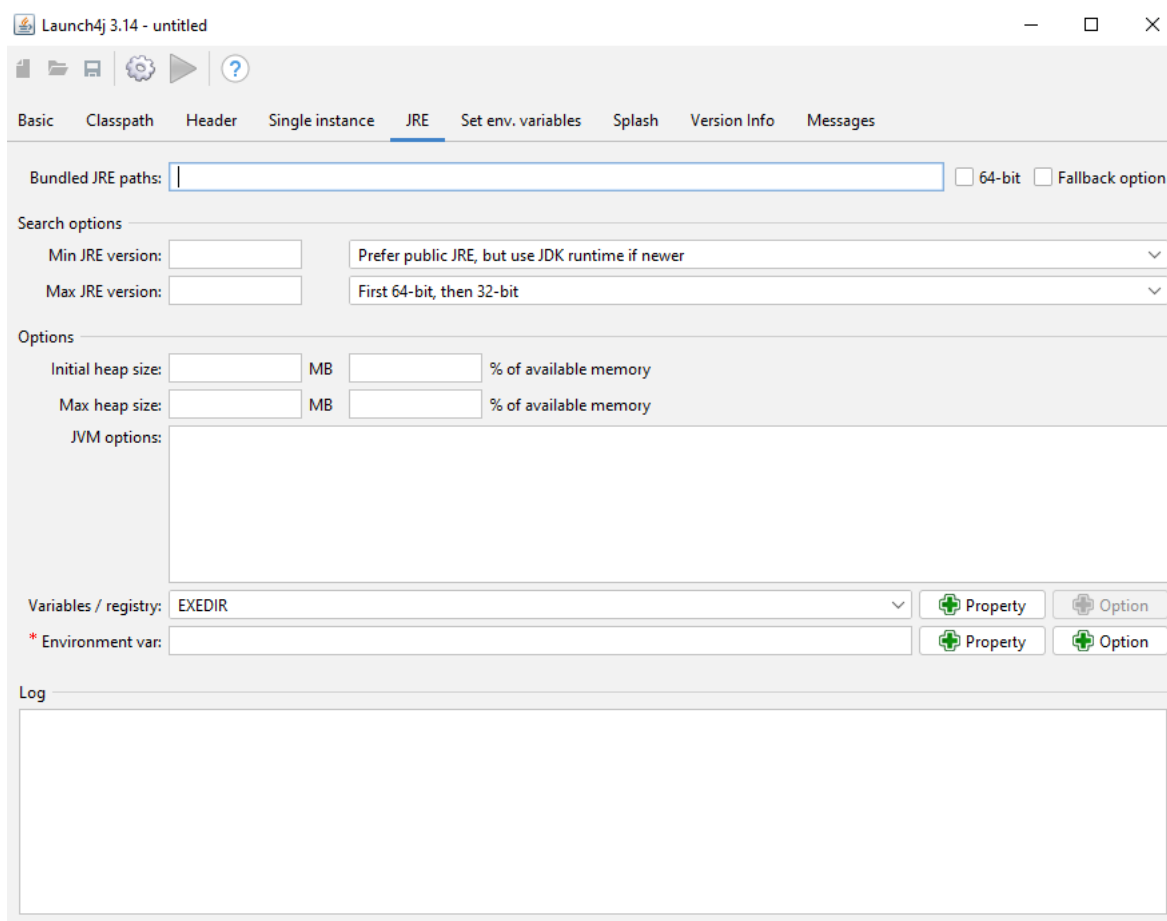
En el caso de la segunda (Jar), indicaremos cuál es nuestro archivo jar, generado en el apartado anterior.

De manera opcional podemos indicar en la opción Icon un archivo en formato .ico que nos servirá como icono del ejecutable.

El resto de las opciones que aparecen en esta pestaña se mantendrán con sus valores por defecto.

A continuación, pasaremos a la pestaña JRE.

Dada la necesidad de que cualquier usuario pueda ejecutar nuestra aplicación sin tener instalado Java en sus equipos o no disponer de la versión concreta para su funcionamiento, deberemos añadir Java al ejecutable para que, al acceder a la aplicación, esta se ejecute correctamente.



5 Pestaña JRE en Launch4j

Una vez en la pestaña JRE, deberemos configurar las opciones siguientes:

- **Bundled JRE paths:** Indicaremos la ruta de la carpeta JRE que queremos incluir en nuestro ejecutable.
- **Min JRE version:** Indicaremos la versión exacta de JRE que vamos a instanciar. En su menú desplegable, situado a su derecha, seleccionaremos la opción Only use private JDK runtimes.

Para configurar la carpeta de JRE que queremos instanciar junto a nuestro ejecutable, deberemos acceder a la ruta donde tengamos instalado JAVA en nuestro equipo, generalmente se encontrará en C:\Program Files (x86)\Java\jre1.8.0_271.

Nombre	Fecha de modificación	Tipo	Tamaño
bin	08/11/2020 17:50	Carpeta de archivos	
legal	08/11/2020 17:50	Carpeta de archivos	
lib	08/11/2020 17:50	Carpeta de archivos	
COPYRIGHT	08/11/2020 17:50	Archivo	4 KB
LICENSE	08/11/2020 17:50	Archivo	1 KB
README	08/11/2020 17:50	Documento de te...	1 KB
release	08/11/2020 17:50	Archivo	1 KB
THIRDPARTYLICENSEREADME	08/11/2020 17:50	Documento de te...	1 KB
THIRDPARTYLICENSEREADME-JAVAFX	08/11/2020 17:50	Documento de te...	1 KB
Welcome	08/11/2020 17:50	Microsoft Edge H...	1 KB

6 Contenido carpeta JRE del Sistema

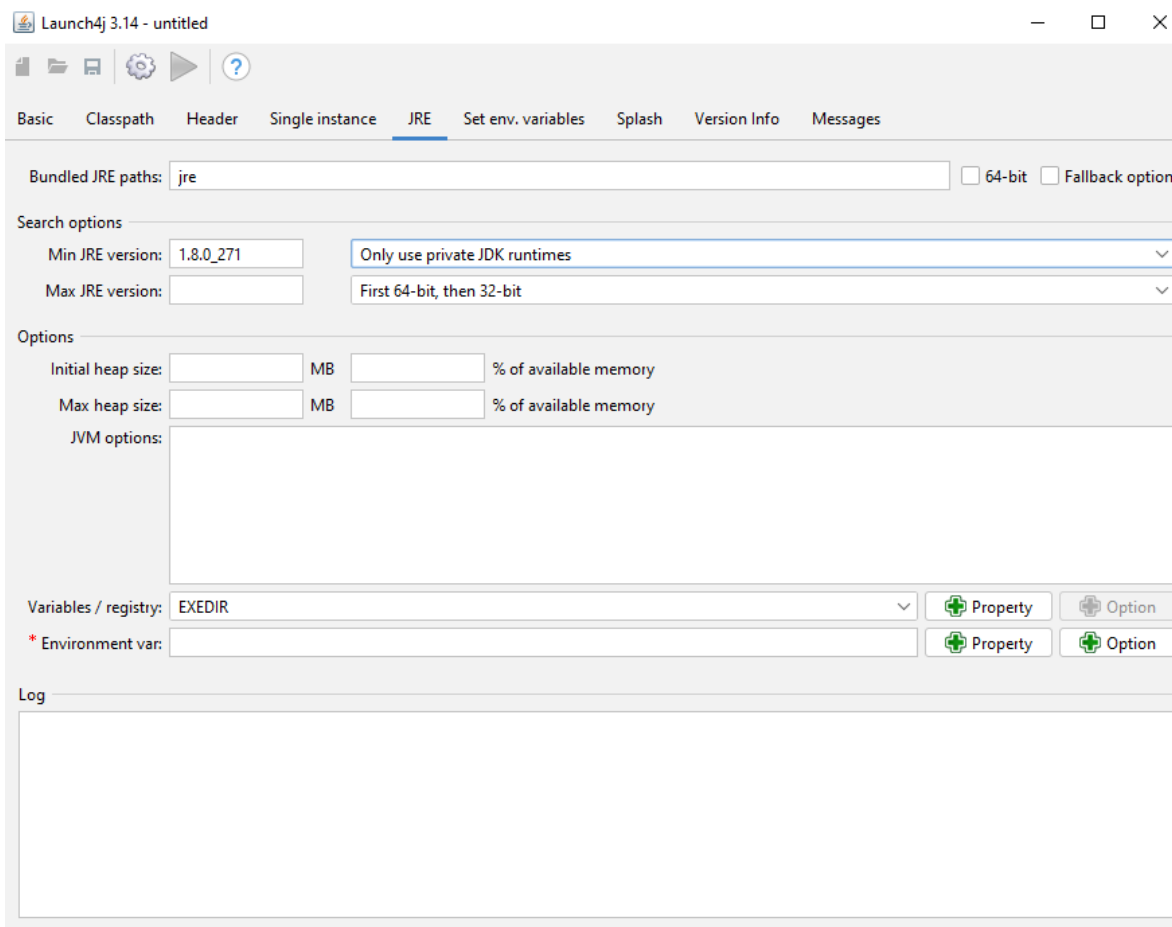
Una vez dentro, copiaremos únicamente las subcarpetas bin y lib. Con ellas crearemos una nueva carpeta donde las pegaremos.

Nombre	Fecha de modificación	Tipo	Tamaño
bin	10/04/2021 12:19	Carpeta de archivos	
lib	10/04/2021 12:19	Carpeta de archivos	

7 Carpeta con los archivos necesarios JRE

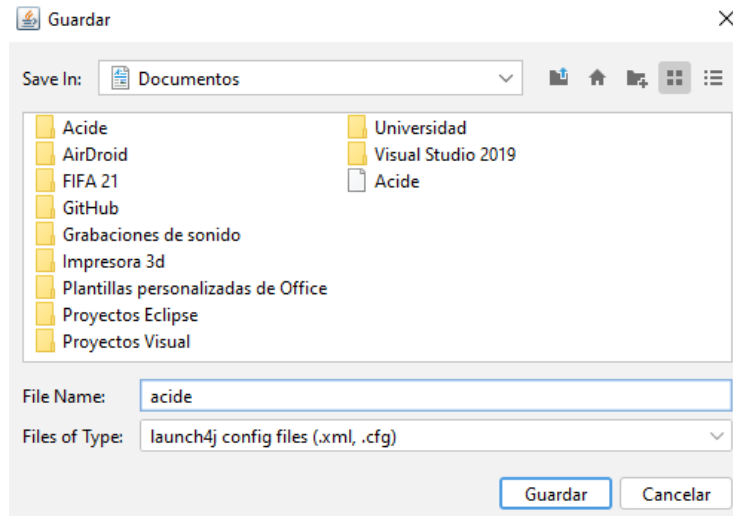
Por último, indicaremos el nombre de esta nueva carpeta en la opción Bundled JRE paths. Más adelante incluiremos esta carpeta en el instalador.

A continuación, se muestra una captura ejemplo con una configuración válida para este proceso.



8 Captura con configuración tipo

Para terminar de generar el ejecutable .exe, deberemos pulsar en el botón con forma de tuerca presente en la toolbar del programa. Se nos abrirá un diálogo para guardar el script que genera la aplicación al crear el archivo .exe.



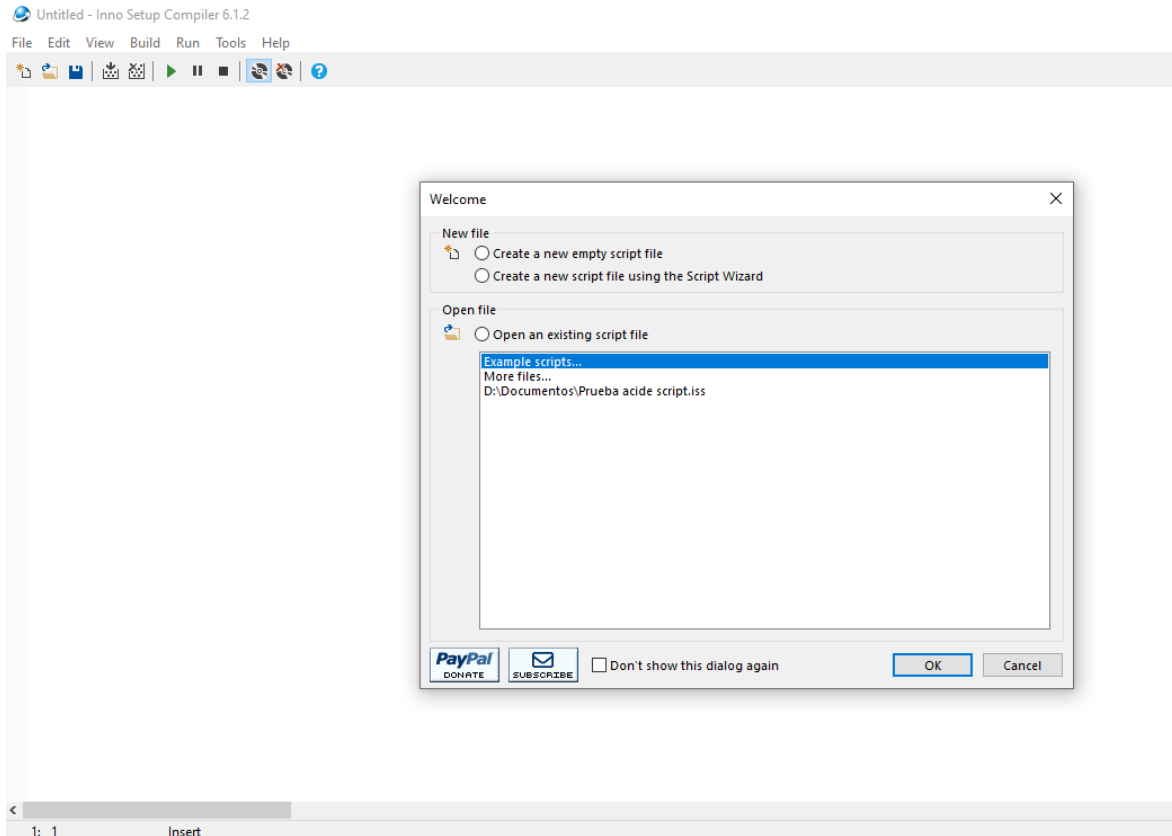
9 Diálogo para guardar el script de creación

Al finalizar, deberíamos tener nuestro archivo .exe en la ruta que especificamos durante el proceso de creación en la opción Output file.

7.3.3 Creación del instalador:

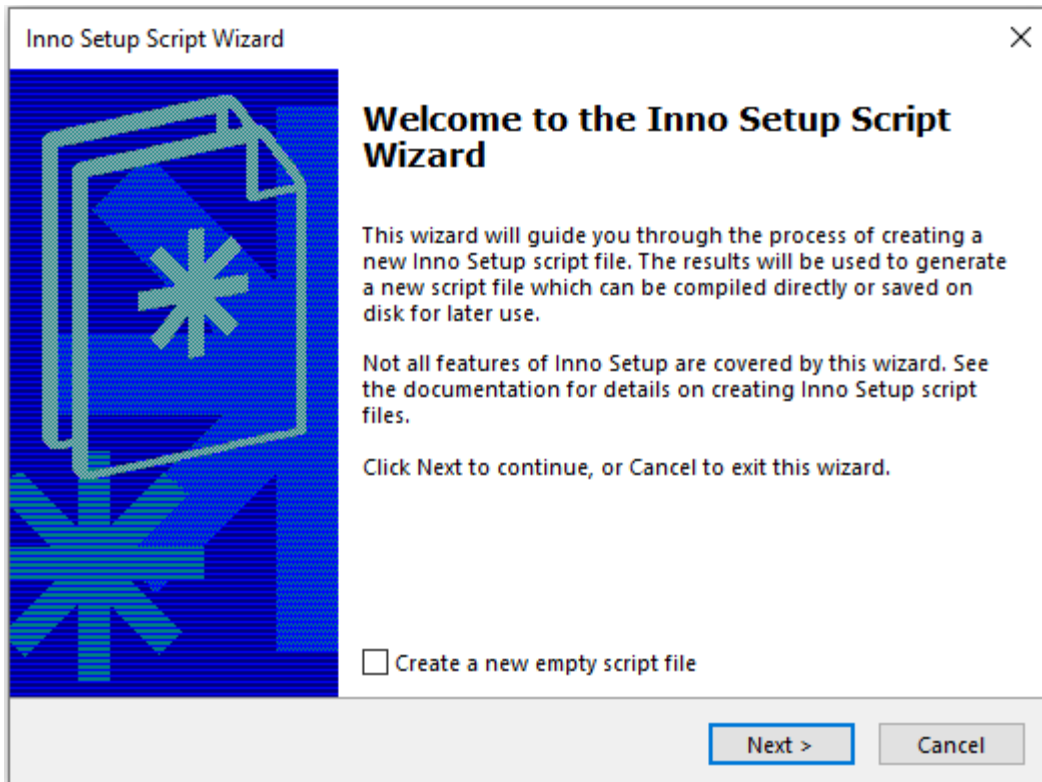
Para este proceso se usará el programa Inno Setup Compiler [4], que nos permite añadir al instalador todas las carpetas y archivos necesarios para la correcta instalación de nuestra aplicación.

Una vez abierto el programa observamos la siguiente ventana:




10 Vista principal Inno Setup Compiler

Deberemos escoger la opción “create a new script file using the Script Wizard”. Tras esto se nos abrirá el asistente de creación del instalador, donde daremos en Next.




11 Vista principal asistente creación del instalador

A continuación, deberemos rellenar datos referentes a nuestro programa, como son el nombre y la versión del mismo. Los campos de creador y web pueden dejarse en blanco.

Inno Setup Script Wizard 

Application Information
Please specify some basic information about your application.



Application name:

Application version:

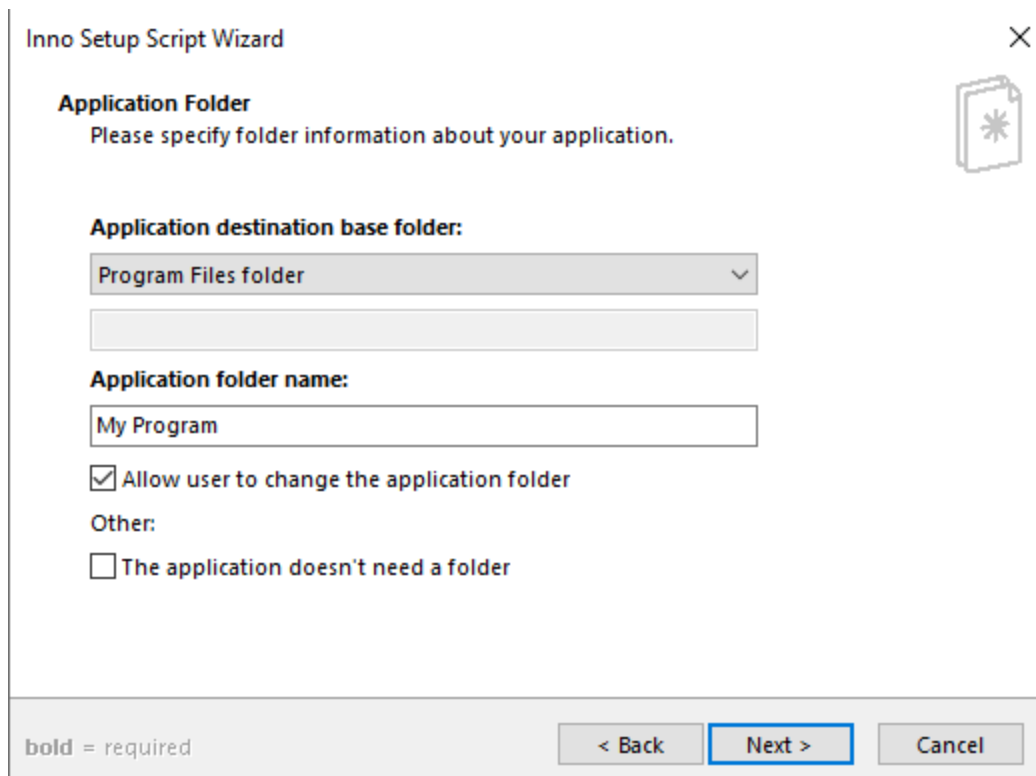
Application publisher:

Application website:

bold = required

12 Información de la aplicación

El siguiente paso nos solicita el directorio en el que queremos que se instale nuestro programa; dejaremos la opción por defecto de programe files folder, aunque antes de finalizar el proceso lo cambiaremos. También deberemos indicar si nuestro programa precisa de una carpeta donde alojar sus archivos, así como el nombre de esta.



The screenshot shows the 'Inno Setup Script Wizard' window, specifically the 'Application Folder' step. The window has a title bar with a close button (X) and a help icon (question mark). The main content area is titled 'Application Folder' and contains the instruction 'Please specify folder information about your application.' Below this, there are three sections: 'Application destination base folder:' with a dropdown menu set to 'Program Files folder'; 'Application folder name:' with a text box containing 'My Program'; and a checkbox labeled 'Allow user to change the application folder' which is checked. Below the checkbox is the text 'Other:' followed by an unchecked checkbox labeled 'The application doesn't need a folder'. At the bottom of the window, there is a status bar with the text 'bold = required' and three buttons: '< Back', 'Next >' (highlighted with a blue border), and 'Cancel'.

Inno Setup Script Wizard

Application Folder
Please specify folder information about your application.

Application destination base folder:
Program Files folder

Application folder name:
My Program

☒ Allow user to change the application folder

Other:
☐ The application doesn't need a folder

bold = required < Back Next > Cancel

13 Configuración para alojar nuestro programa

Al hacer click en Next, pasaremos a la ventana donde deberemos indicar cuál es el archivo ejecutable de nuestra aplicación, el que hemos generado en el apartado anterior, así como todos los archivos y carpetas necesarios para el correcto funcionamiento de nuestro programa.

El resto de las opciones que aparecen las dejaremos con sus valores por defecto.

Inno Setup Script Wizard

Application Files
Please specify the files that are part of your application.

Application main executable file:

D:\Descargas\Prueba tfg\acide.exe Browse...

☒ Allow user to start the application after Setup has finished

☐ The application doesn't have a main executable file

Other application files:

D:\Descargas\p*
D:\Descargas\Prueba tfg\des.exe
D:\Descargas\Prueba tfg\DES6.6.acideProject
D:\Descargas\Prueba tfg\MSDOS.acideProject
D:\Descargas\Prueba tfg\test.bat

Add file(s)...
Add folder...
Edit...
Remove

bold = required < Back Next > Cancel

14 Configuración archivos necesarios del programa

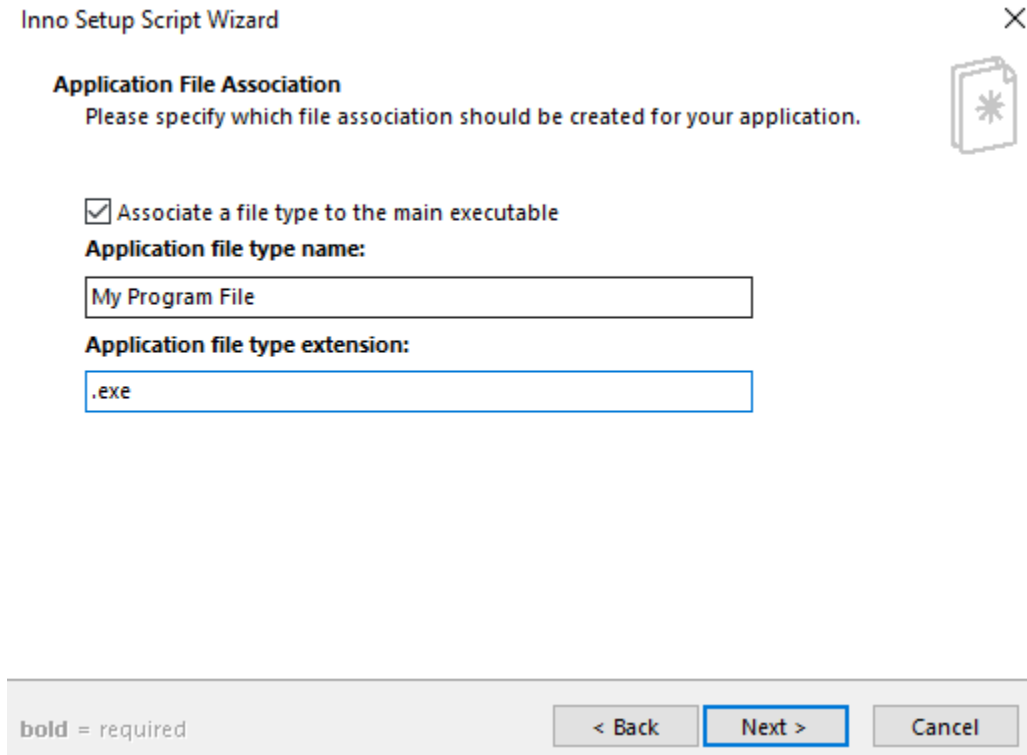
Si nuestra aplicación necesita carpetas de archivos, será necesario englobar todas ellas en una carpeta; si no se procede de este modo y las importamos carpeta a carpeta, Inno Setup Compiler [4] importará los archivos que haya en ellas, pero no la propia carpeta.

Se muestra un ejemplo de cómo proceder con las carpetas necesarias para el instalador de ACIDE.

Nombre	Fecha de modificación	Tipo	Tamaño
configuration	10/04/2021 19:19	Carpeta de archivos	
examples	09/04/2021 20:44	Carpeta de archivos	
jre	10/04/2021 12:19	Carpeta de archivos	
language	09/04/2021 17:22	Carpeta de archivos	
resources	09/04/2021 17:22	Carpeta de archivos	

15 Contenido carpeta p incluida en el instalador

Al pulsar en Next deberemos indicar la extensión de nuestro instalador, así como el nombre que le queramos dar al mismo. La extensión que deberemos marcar será .exe. La opción seleccionada la dejaremos por defecto.



The screenshot shows the 'Inno Setup Script Wizard' window, specifically the 'Application File Association' step. The window has a title bar with a close button (X) and a help icon (question mark). The main text reads: 'Application File Association' followed by 'Please specify which file association should be created for your application.' Below this, there is a checkbox labeled 'Associate a file type to the main executable' which is checked. Underneath the checkbox, there are two labels: 'Application file type name:' and 'Application file type extension:'. The 'Application file type name' field contains the text 'My Program File'. The 'Application file type extension' field contains the text '.exe'. At the bottom of the window, there is a status bar that says 'bold = required'. To the right of the status bar are three buttons: '< Back', 'Next >', and 'Cancel'. The 'Next >' button is highlighted with a blue border.

Inno Setup Script Wizard

Application File Association
Please specify which file association should be created for your application.

☒ Associate a file type to the main executable

Application file type name:
My Program File

Application file type extension:
.exe

bold = required < Back Next > Cancel

16 Extensión y nombre del instalador

Al pasar a la siguiente ventana, nos permitirá decidir sobre crear accesos en el menú de inicio para el programa, así como permitir que el usuario genere accesos directos del mismo.

Application Shortcuts

Please specify which shortcuts should be created for your application.

☒ **Create a shortcut to the main executable in the common Start Menu Programs folder**

Application Start Menu folder name:

My Program

☒ Allow user to change the Start Menu folder name☐ Allow user to disable Start Menu folder creation☐ Create an Internet shortcut in the Start Menu folder☐ Create an Uninstall shortcut in the Start Menu folder**Other shortcuts to the main executable:**☒ Allow user to create a desktop shortcut

bold = required

< Back


Next >

Cancel

17 Accesos directos y menú de inicio

En el siguiente paso deberemos indicar si nuestro programa dispone de archivos de licencia, para que el usuario pueda aceptarlos si desea instalarlo. También podremos añadir cualquier documento informativo de nuestra aplicación que aparecerá a la hora de abrir el instalador o al finalizar el proceso. Si no tenemos la necesidad de incluir ninguno podremos pasar a la siguiente pantalla.

Inno Setup Script Wizard ✕

Application Documentation 

Please specify which documentation files should be shown by Setup during installation.

License file:

Browse...

Information file shown before installation:

Browse...

Information file shown after installation:

Browse...

< Back

Next >

Cancel

18 Licencias y documentos informativos

A continuación, deberemos indicar para qué usuario del sistema se instala nuestra aplicación. Para la configuración de ACIDE se dejaron las opciones marcadas por defecto.

Inno Setup Script Wizard



Setup Install Mode

Please specify in which install mode Setup should run.



Install mode:

- ☒ Administrative install mode (install for all users)
- ☐ Non administrative install mode (install for current user only)
- ☐ Allow user to override the install mode via the command line
- ☐ Ask the user to choose the install mode at startup

bold = required

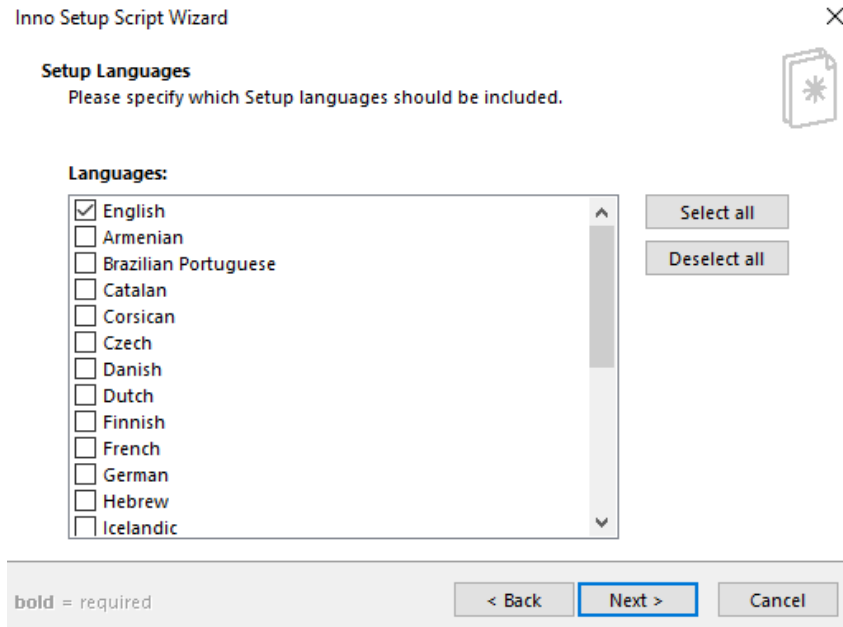
< Back

Next >

Cancel

19 Usuarios con acceso al programa

Al pulsar en Next pasaremos a la ventana que nos permitirá seleccionar los idiomas que queremos que tenga nuestro instalador.



20 Idiomas del instalador

Por último, deberemos indicar la ruta donde queremos que se guarde nuestro instalador, el nombre que queramos asignarle y, si así lo deseamos, el icono del mismo y una contraseña para iniciarlo.

Inno Setup Script Wizard

Compiler Settings

Please specify some basic compiler settings.



Custom compiler output folder:

Browse...

Compiler output base file name:

Custom Setup icon file:

Browse...

Setup password:

< Back

Next >

Cancel

21 Nombre setup, icono y ruta de creación

Finalmente pulsaremos next y finish para que nos genere el script con toda la configuración. Cuando nos aparezca el diálogo para compilarlo ahora diremos que no y pasaremos a añadir las últimas configuraciones al script.

Primero deberemos cambiar la propiedad DefaultDirName dentro del apartado setup del script por la siguiente línea: DefaultDirName={sd}\{#MyAppName}

Esto permitirá que nuestra aplicación se instale en la carpeta raíz del disco del sistema. Por ejemplo C:\ACIDE

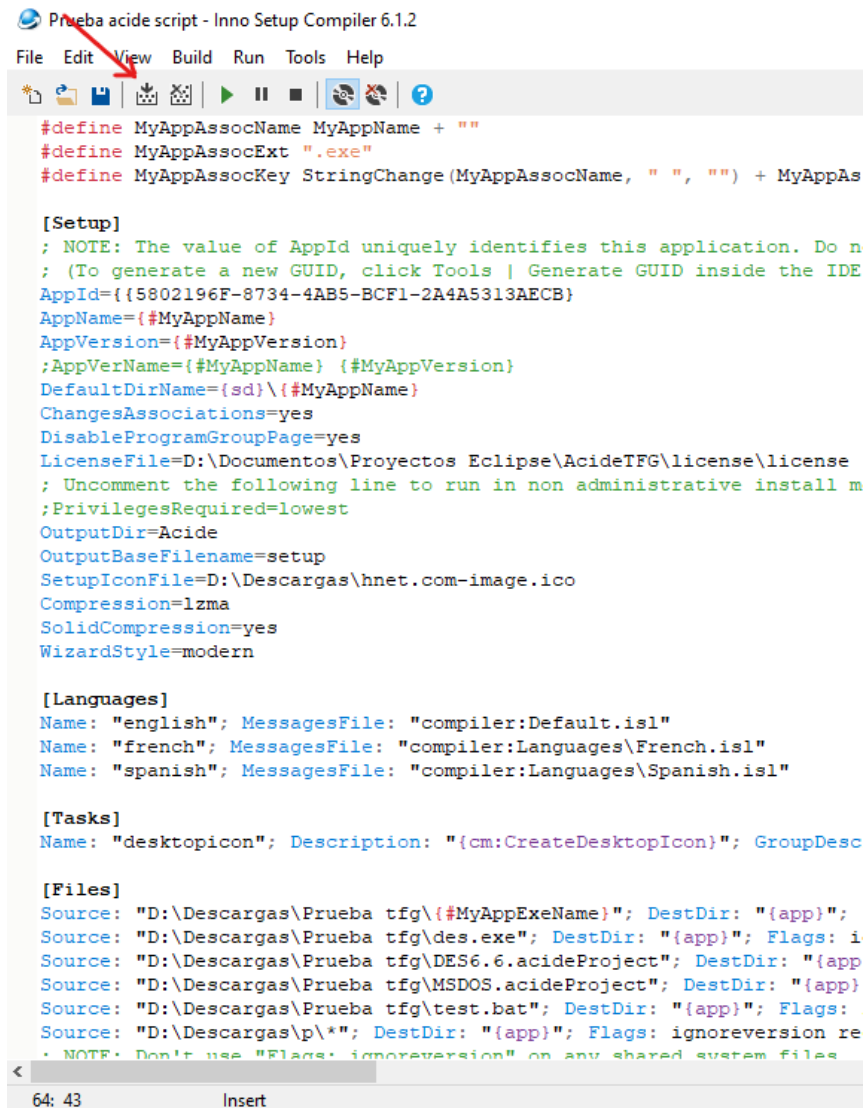
Por último, deberemos añadir un nuevo bloque al script que hará que al desinstalar nuestra aplicación se borren todos los datos de la misma.

[UninstallDelete]

Type: files; Name: "{app}*";

Type: filesandordirs; Name: "{app}"

Con todo esto, podemos pulsar en el botón para compilar y generar nuestro instalador. Una vez finaliza el proceso tendremos nuestro instalador en la ruta que le indicamos durante el proceso de creación.



```
#define MyAppAssocName MyAppName + ""
#define MyAppAssocExt ".exe"
#define MyAppAssocKey StringChange(MyAppAssocName, " ", "") + MyAppAs

[Setup]
; NOTE: The value of AppId uniquely identifies this application. Do n
; (To generate a new GUID, click Tools | Generate GUID inside the IDE
AppId={{5802196F-8734-4AB5-BCF1-2A4A5313AECB}
AppName={#MyAppName}
AppVersion={#MyAppVersion}
;AppVerName={#MyAppName} {#MyAppVersion}
DefaultDirName={sd}\{#MyAppName}
ChangesAssociations=yes
DisableProgramGroupPage=yes
LicenseFile=D:\Documentos\Proyectos Eclipse\AcideTFG\license\license
; Uncomment the following line to run in non administrative install m
;PrivilegesRequired=lowest
OutputDir=Acide
OutputBaseFilename=setup
SetupIconFile=D:\Descargas\hnet.com-image.ico
Compression=lzma
SolidCompression=yes
WizardStyle=modern

[Languages]
Name: "english"; MessagesFile: "compiler:Default.isl"
Name: "french"; MessagesFile: "compiler:Languages\French.isl"
Name: "spanish"; MessagesFile: "compiler:Languages\Spanish.isl"

[Tasks]
Name: "desktopicon"; Description: "{cm:CreateDesktopIcon}"; GroupDesc

[Files]
Source: "D:\Descargas\Prueba tfg\{#MyAppExeName}"; DestDir: "{app}";
Source: "D:\Descargas\Prueba tfg\des.exe"; DestDir: "{app}"; Flags: i
Source: "D:\Descargas\Prueba tfg\DES6.6.acideProject"; DestDir: "{app
Source: "D:\Descargas\Prueba tfg\MSDOS.acideProject"; DestDir: "{app}
Source: "D:\Descargas\Prueba tfg\test.bat"; DestDir: "{app}"; Flags:
Source: "D:\Descargas\p\*"; DestDir: "{app}"; Flags: ignoreversion re
; NOTE: Don't use "Flags: ignoreversion" on any shared system files
```

22 Compilar y generar el instalador

Apéndice B - Generación automática del instalador para ACIDE

Además de disponer de la opción de generar el instalador de ACIDE haciendo uso de la interfaz que nos proporciona Inno Setup Compiler [4], podemos emplear los comandos que pone a nuestra disposición, agilizando el proceso de creación.

Por ello, según se ha comentado en el apartado correspondiente de este documento, se ha creado un archivo bat que permitirá generar el instalador de ACIDE, evitando largos procesos a través de cuadros de diálogo y configuraciones.

A continuación, se exponen los pasos a seguir para crear este archivo, así como la estructura de directorios necesaria para su correcto funcionamiento.

En primer lugar será necesario consultar los comandos que nos ofrece Inno Setup Compiler [4] para generar los ejecutables, haciendo uso de los scripts de configuración.

En este punto se ha empleado la siguiente orden:

```
"C:\Program Files (x86)\Inno Setup 6\ISCC.exe"
```

Se debe tener en cuenta que la ruta depende del directorio en el que el usuario haya instalado Inno Setup [4] en su sistema.

Una vez tenemos claro este comando, que se encargará de compilar el script con la configuración de ACIDE, paso a explicar la estructura de directorios necesaria para que funcione correctamente.

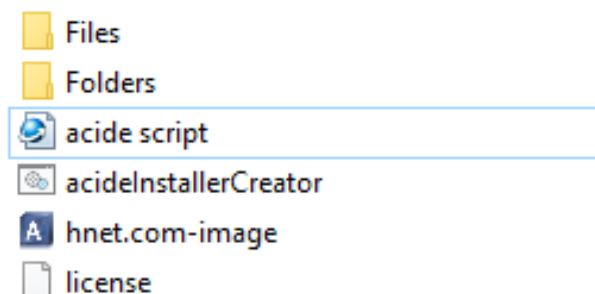


Figura 4-1 Estructura necesaria de la carpeta donde se aloje el bat

- En la carpeta Files deberemos incluir todos los archivos necesarios para el proyecto, así como el ejecutable que hayamos creado con el proceso descrito en el apéndice anterior (7.3.2 Generación del ejecutable .exe:)

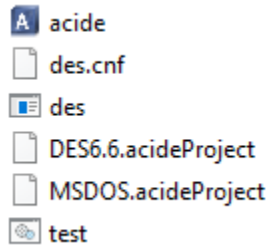


Figura 4-2 Contenido carpeta Files

Si se quieren añadir archivos con nombres distintos o simplemente nuevos archivos, será necesario modificar el script de creación del instalador.

- La carpeta Folders contiene todas las carpetas que necesita ACIDE para poder funcionar. Es decir, carpetas con archivos de configuración dedicados a cada uno de los componentes de ACIDE.

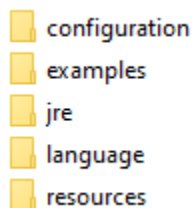


Figura 4-3 Contenido de la carpeta Folders

En este caso, si en versiones posteriores de ACIDE, debido al desarrollo de nuevas funcionalidades, fuese preciso añadir alguna carpeta más, únicamente sería necesario alojarla en este directorio, sin necesidad de modificar nada más.

- Acide script: hace referencia al script que se compilará gracias al archivo bat y que dará lugar al instalador de ACIDE. A continuación, se explican las secciones más importantes del archivo.

```
[Files]
Source: ".\Files\{MyAppExeName}"; DestDir: "{app}"; Flags: ignoreversion
Source: ".\Files\des.exe"; DestDir: "{app}"; Flags: ignoreversion;
Source: ".\Files\DES6.6.acideProject"; DestDir: "{app}"; Flags: ignoreversion
Source: ".\Files\MSDOS.acideProject"; DestDir: "{app}"; Flags: ignoreversion
Source: ".\Files\test.bat"; DestDir: "{app}"; Flags: ignoreversion
Source: ".\Files\des.cnf"; DestDir: "{app}"; Flags: ignoreversion
Source: ".\Folders\*"; DestDir: "{app}"; Flags: ignoreversion recursesubdirs createallsubdirs;
; NOTE: Don't use "Flags: ignoreversion" on any shared system files
```

Figura 4-4 Sección Files donde registrar todos los archivos que se quieran añadir

```
[UninstallDelete]
Type: files; Name: "{app}\*";
Type: filesandordirs; Name: "{app}"
```

Figura 4-5 Sección para borrar cualquier archivo de ACIDE al desinstalar

- hnet.com-image: se corresponde con el icono para el setup, en formato ico.
- license: archivo de licencia de ACIDE, que el usuario deberá aceptar durante el proceso de instalación.
- acideInstallerCreator: archivo bat para automatizar el proceso de creación del instalador de ACIDE. Su contenido se muestra en la siguiente captura.
-

```
@echo_off
echo Generating Acide installer
"C:\Program Files (x86)\Inno Setup 6\ISCC.exe" ".\acide script.iss"
start %windir%\explorer.exe Setup
```

Figura 4-6 Contenido archivo bat

Las sentencias del archivo, por orden de declaración, realizan:

- Muestra por pantalla el texto "Generating Acide installer"
- Compila el script y genera el instalador de ACIDE
- Abre el explorador en la ruta donde se ha generado el instalador de ACIDE.